



# Oracle Taleo Enterprise Edition

---

## Using Connect Client

**21B**



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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

|                             |            |
|-----------------------------|------------|
| <b>Preface</b>              | <b>i</b>   |
| <b>1 Introduction</b>       | <b>1</b>   |
| Taleo Connect               | 1          |
| <b>2 Installation</b>       | <b>3</b>   |
| Files and Folders           | 3          |
| In Windows                  | 5          |
| In Unix/Linux               | 7          |
| <b>3 Description</b>        | <b>11</b>  |
| Taleo Connect Client Basics | 11         |
| Default Configuration Board | 24         |
| Import                      | 29         |
| Export                      | 40         |
| Configuration               | 53         |
| Runtime                     | 79         |
| Console                     | 80         |
| Command Line                | 84         |
| <b>4 Getting Started</b>    | <b>89</b>  |
| Starting                    | 89         |
| Importing                   | 91         |
| Exporting                   | 95         |
| <b>5 Using TCC</b>          | <b>99</b>  |
| Hierarchical Entities       | 99         |
| Filters                     | 113        |
| Miscellaneous               | 115        |
| <b>6 Net Change</b>         | <b>121</b> |
| Description                 | 121        |

---

|                     |     |
|---------------------|-----|
| Import Files        | 125 |
| Configuration Files | 140 |
| Using Net Change    | 144 |

## **7 Templates** **153**

---

|                                  |     |
|----------------------------------|-----|
| Best Practice Templates          | 153 |
| Candidate Standard Net Change    | 153 |
| Candidate Standard Export        | 155 |
| Department Standard Net Change   | 155 |
| JobField Standard Net Change     | 156 |
| JobTemplate Standard Net Change  | 157 |
| Location Standard Net Change     | 159 |
| LUDS Standard Net Change         | 159 |
| Organization Standard Net Change | 160 |
| Position Standard Net Change     | 161 |
| Requisition Standard Merge       | 161 |
| User Standard Net Change         | 164 |

## **8 Advanced Queries** **167**

---

|                           |     |
|---------------------------|-----|
| Introduction              | 167 |
| Inserting Complex Queries | 167 |
| Examples                  | 168 |
| Oracle Reserved Words     | 189 |

## **9 Security** **193**

---

|                                 |     |
|---------------------------------|-----|
| Data Confidentiality            | 193 |
| User Authentication             | 193 |
| Stored Passwords                | 193 |
| Specifying Integration Password | 196 |
| Override ENCRYPTED_PASSWORD     | 197 |
| HTTP Proxy                      | 197 |
| Encrypting Proxy Password       | 198 |
| Microsoft NTLM Support          | 199 |
| Data at Rest                    | 200 |

## **10 Appendix** **201**

---

|            |     |
|------------|-----|
| Appendix A | 201 |
|------------|-----|

Appendix B  
Appendix C

203  
204



# Preface

This preface introduces information sources that can help you use the application and this guide.

## Oracle Applications Guides

To find guides for Oracle Applications, go to the Oracle Help Center at <http://docs.oracle.com/>.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the [Oracle Accessibility Program website](#).

## Comments and Suggestions

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit [My Oracle Support](#) or visit [Accessible Oracle Support](#) if you are hearing impaired.

### Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: [talent\\_acquisition\\_doc\\_feedback\\_ww\\_grp@oracle.com](mailto:talent_acquisition_doc_feedback_ww_grp@oracle.com).





# 1 Introduction

## Taleo Connect

### Taleo Connect Server

Taleo Connect Server is the server-side portion of the Taleo Connect Platform embedded within the Oracle Taleo Enterprise Edition 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 Oracle Taleo Enterprise Edition 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 (TCC) is the client-side portion of the Taleo Connect Platform that resides in the customer's environment. It enables customers to build and run 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

TCC requires a user account to be able to access the zone and conduct integration jobs. This user needs to be configured with System Integrator permissions for the products that you wish to integrate with.

It is recommended that when creating integration user accounts to be used with TCC, that different UID/PWD combinations are used for each zone. This will remove any inadvertent issues around zone access.

# 2 Installation

## 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\_Id.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

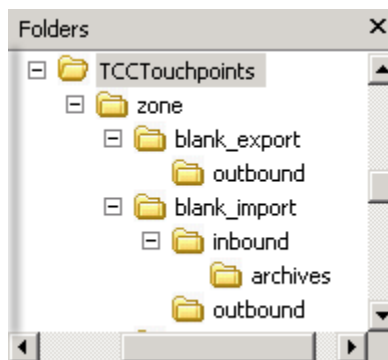
Configuration file: newhire\_export\_cfg.xml

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.

The image illustrates a well organized folder structure.



### Level 0

This is the root folder for Taleo integrations. It contains the following:

- Zones

A folder for each zone that is available such as production, staging, development.

## Level 1

This is the Zones folder level. It contains the following:

- touchpoint\_action  
A folder for each import and export integration point.

## Level 2

This is the integration point folder level. It contains the following:

- 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. It contains the following:

- archives  
The archives folder to save imported data files.

# In Windows

## Installing TCC in Windows

One of the following operating systems pre-installed:

- Windows Server 2008
- Windows Server 2012
- Windows Server 2016
- Linux

Hardware:

- Design and development workstation, ~1GHz, 256 MB RAM
- Large volume production workstation, ~1 GHz, 512MB RAM

Software:

- Product Integration Pack (PIP) setup program

- Taleo Connect Client setup program

Security approval from your information security department to install TCC.

TCC installation packages come bundled with a 32 bit version of Java Runtime Environment.

- 17.0 - JRE 1.8

TaleoConnectClient.exe always uses the information in the jre folder created during installation, no matter what value is specified in JAVA\_HOME. If the jre folder is deleted or renamed, making it impossible to find, the value set in JAVA\_HOME is used. If JAVA\_HOME is set to a 32 bit version everything works. If JAVA\_HOME is set to a 64 bit version the "Can't load IA 32-bit .dll on a AMD 64-bit platform" error is displayed.

TaleoConnectClient.bat will work with 32 or 64 bit versions because there is no UI to display. The presence or absence of the jre folder, or the value set in JAVA\_HOME have no impact.

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.
6. 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.

- 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.

1. Launch Taleo Connect Client.  
If you have selected a folder containing at least one valid PIP file during installation, the Ping the Taleo Product window opens. Otherwise, the No product integration pack file configured window opens. Navigate to the ProductIntegrationPack folder and select it. The Ping the Taleo Product window opens.
2. Select the product to connect to from the pull down menu.
3. 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.

4. Configure the authentication information.
  - a. Select authentication.
  - b. Enter user name.
  - c. Enter password.
5. Click Ping. The Ping the Taleo product window closes. An information window indicating the creation of a new endpoint opens.
6. Click OK.

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. Oracle recommends following the upgrade process described below to ensure smooth transition and minimal impact on existing integration scripts.

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

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 Net Change, TCC users MUST run the “Net Change Setup” to create the Net Change 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 Linux enables users to run import and export configuration files, in a command line environment.

- Operating system:
  - Linux
- Hardware:

Large volume production workstation, ~1 GHz, 512MB RAM

- Software:

JRE 1.8.0 or JRE 8.0

TaleoConnectClient-X.Y.Z.tar.gz file

- Security approval from your information security department to install TCC.

1. Copy the TaleoConnectClient-[version].tar.gz file to your 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`.

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

## Testing TCC in Unix/Linux

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

- TCC is installed in Linux.
  - Production and staging zones are up and running.
  - Linux system has access to zones.
  - Valid import specification and data file, and associated configuration file.
  - Valid export specification and associated configuration file.
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. Oracle recommends following the upgrade process described below to ensure smooth transition and minimal impact on existing integration scripts.

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

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 Net Change, TCC users MUST run the “Net Change Setup” to create the Net Change database for the new TCC version.
6. Modify any existing reference to the TCC installation location to point to the new location.



# 3 Description

## Taleo Connect Client Basics

### Ping Dialog

The Ping the Taleo product window 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**  
Provide the connectivity information to access the product

Product : Recruiting 15B  
Company Code :  
Endpoint :

**Endpoint**  
Configure the connectivity information

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

**Authentication**  
Configure the authentication information

No authentication  
 Basic authentication  
User : johndoe  
Password : ●●●●●●

Cert Based authentication  
User :  
Password :

[Import Endpoint](#) [Certificate Management](#)

Ping

## Connectivity Information

The Connectivity Information section provides the connectivity information to access the product.

- Product: The product to ping is selected from the pull down menu.
- Company Code: The company code is not used.
- 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

The Authentication section provides the authentication context for the requests.

- **No authentication:** No authentication is listed as an option for testing purposes.
- **Basic authentication:** This is the minimum requirement to connect to the Taleo environment.
- **Cert Based authentication:** This is used to sign the requests and responses between Connect Client and Oracle Taleo Enterprise Cloud Service. This is in addition to basic authentication.

For both types of authentication, the user must provide the following information:

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

## Import Endpoint

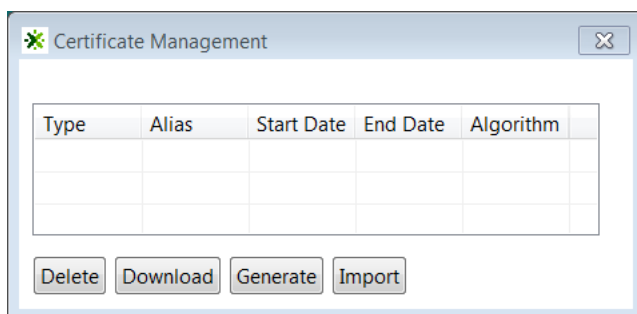
The Import Endpoint link opens a window where the user can locate the original default configuration file supplied by Oracle Support or a legacy environment properties file and retrieve the endpoint settings.

## Certificate Management

The Certificate Management link opens a window where the user can generate, import, download, and delete certificates.

Oracle supports the signing of requests and responses between Taleo Connect Client (TCC) and the Oracle Taleo Enterprise Cloud Service zone. To activate communication signing, a certificate must be generated in TCC and imported in the Taleo zone. A certificate must also be generated in the Taleo zone and imported in TCC. Both certificates are needed for the signing feature to work. When requests and responses are sent by both TCC and Taleo, they are signed by the sender. The signature is validated by the receiver before processing the payload.

When generating a Connect Client certificate, you must set the certificate's validity start date and end date. An alias is also required to identify the certificate. Only one certificate can be valid at a given time. You must create validity periods that do not overlap. Once the certificate is generated, you download it. To complete the certificate setup, the Taleo administrator accesses the TCC Security feature in Taleo and imports the TCC certificate. For instructions on how to import the TCC certificate in Taleo, see the Recruiting User Guide.



## Ping

The Ping button starts the process. When the process is successful, TCC starts. If there is a 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  
Product Integration Packs root folder\*: The path of the root folder where the product Integration Packs are installed. This value is saved in the TaleoConnectClient.ini file.  
\*Take note that system property will override the TaleoConnectClient.ini property.
  - 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 Oracle Support.

- o Client>Runtime

Monitoring auto-refresh rate (seconds): Advanced setting, do not change value unless directed by Oracle Support.

Number of files to keep in monitoring history: Advanced setting, do not change value unless directed by Oracle Support.

- o Console

Advanced settings, do not change values unless directed by Oracle Support.

- o General

Advanced settings, do not change values unless directed by Oracle Support.

- o Help

Advanced settings, do not change values unless directed by Oracle Support.

- o Restore Defaults

Restores all setting to their default values.

- o Apply

Applies new settings to Taleo Connect Client.

## Help

- Help Contents

Opens the online help window.

- About Taleo Connect Client

- o Feature Details: Opens a list of TCC features.

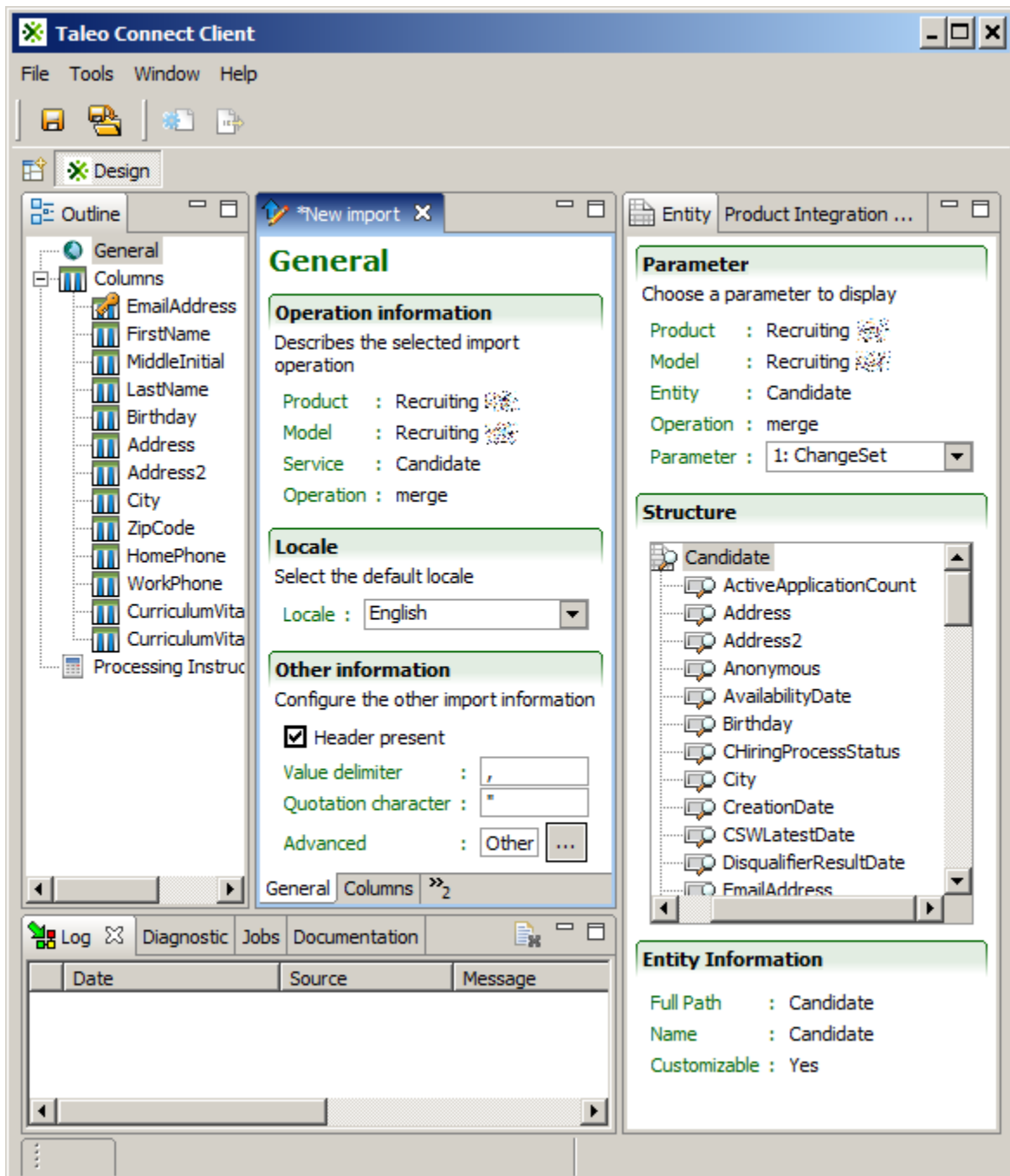
- o Plug-in Details: Opens a list of installed plug-ins.

- o Configuration Details: Displays the TCC configuration details.

## Design Perspective

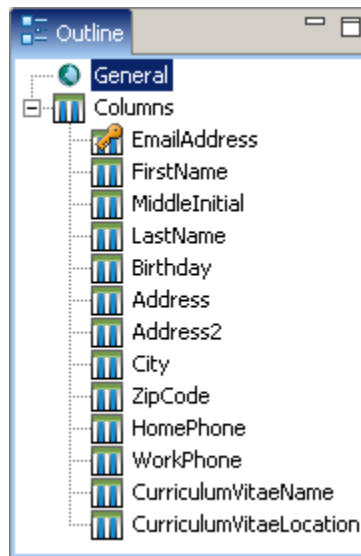
The design perspective is where the integration process workflow is built and where you determine how to extract information from or upload information to the Taleo environment. A specific editor is offered for each of these activities.





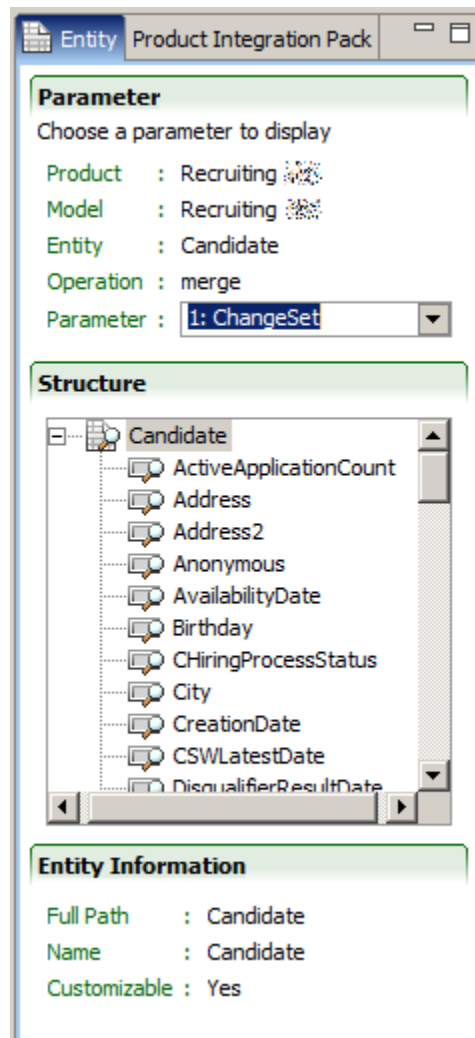
## Outline View

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

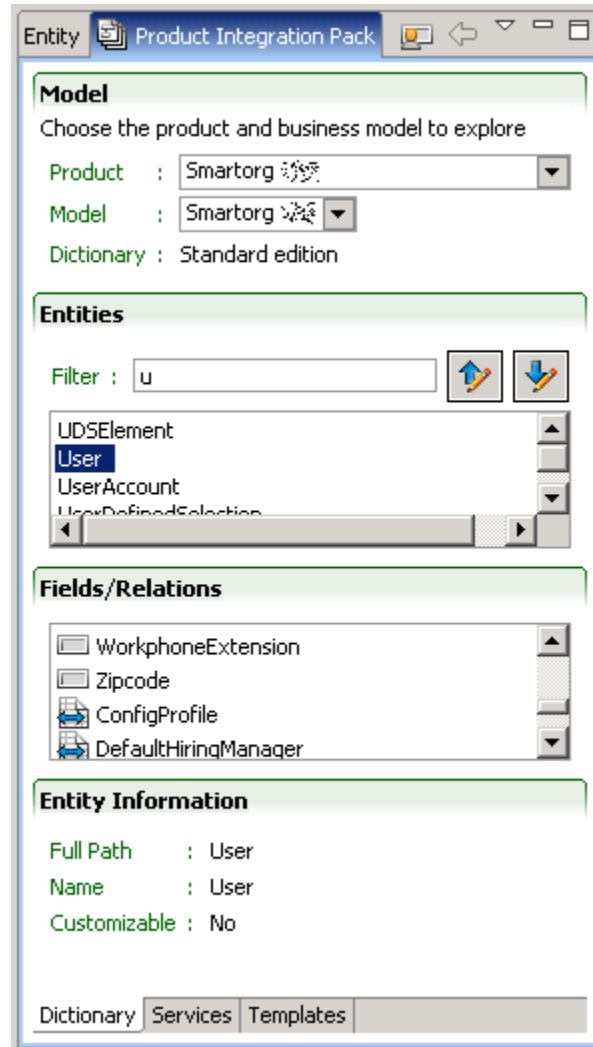
The Product Integration Pack View shows the entire integration offering of a Taleo product with the available standard content.

### 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.

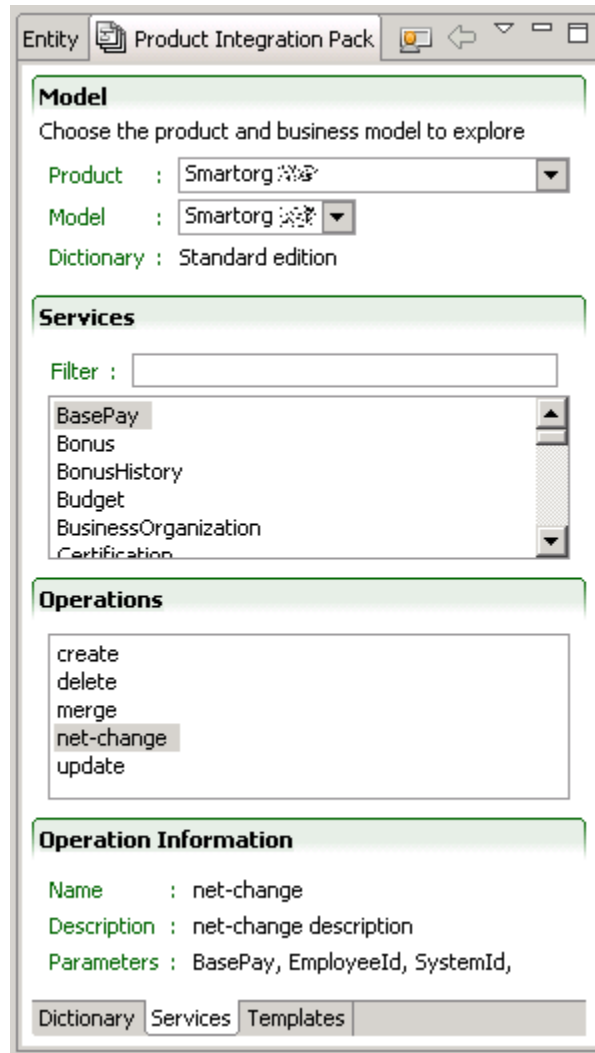
## Dictionary tab



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.

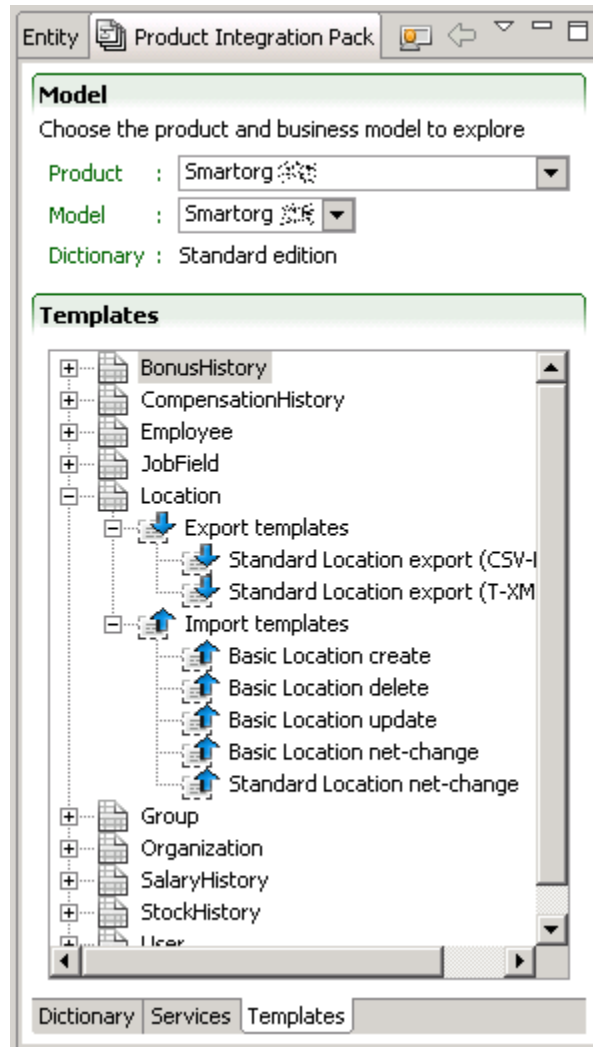
Each entity has a set of fields and relations. The relationships can be navigated to the related entity by double clicking, and back to the original entity with the Return icon.

## 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



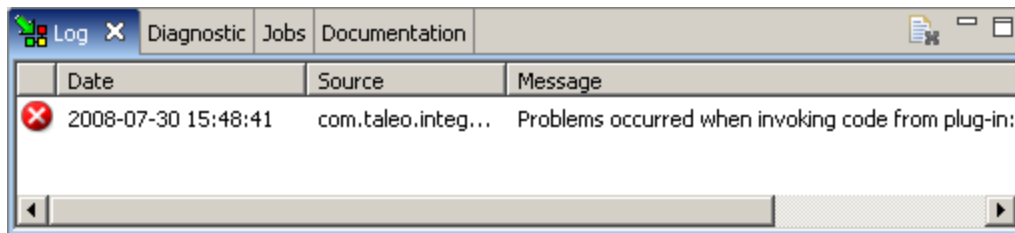
An import template is used to create an import specification for the given entity. The columns provided in the template are typical of integration processes with the Taleo product.

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

## Diagnostic View

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

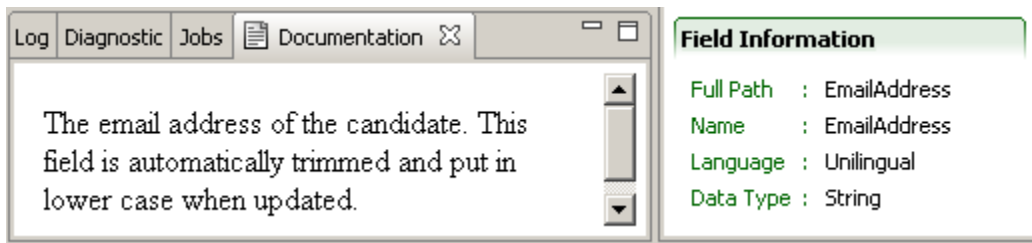
- The first column displays the type of Editor that generated the diagnostic message with an icon.
- The second column displays the severity of the message with an icon.
- The File column displays the file name of the Editor that generated the message.
- The Message column displays the actual description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

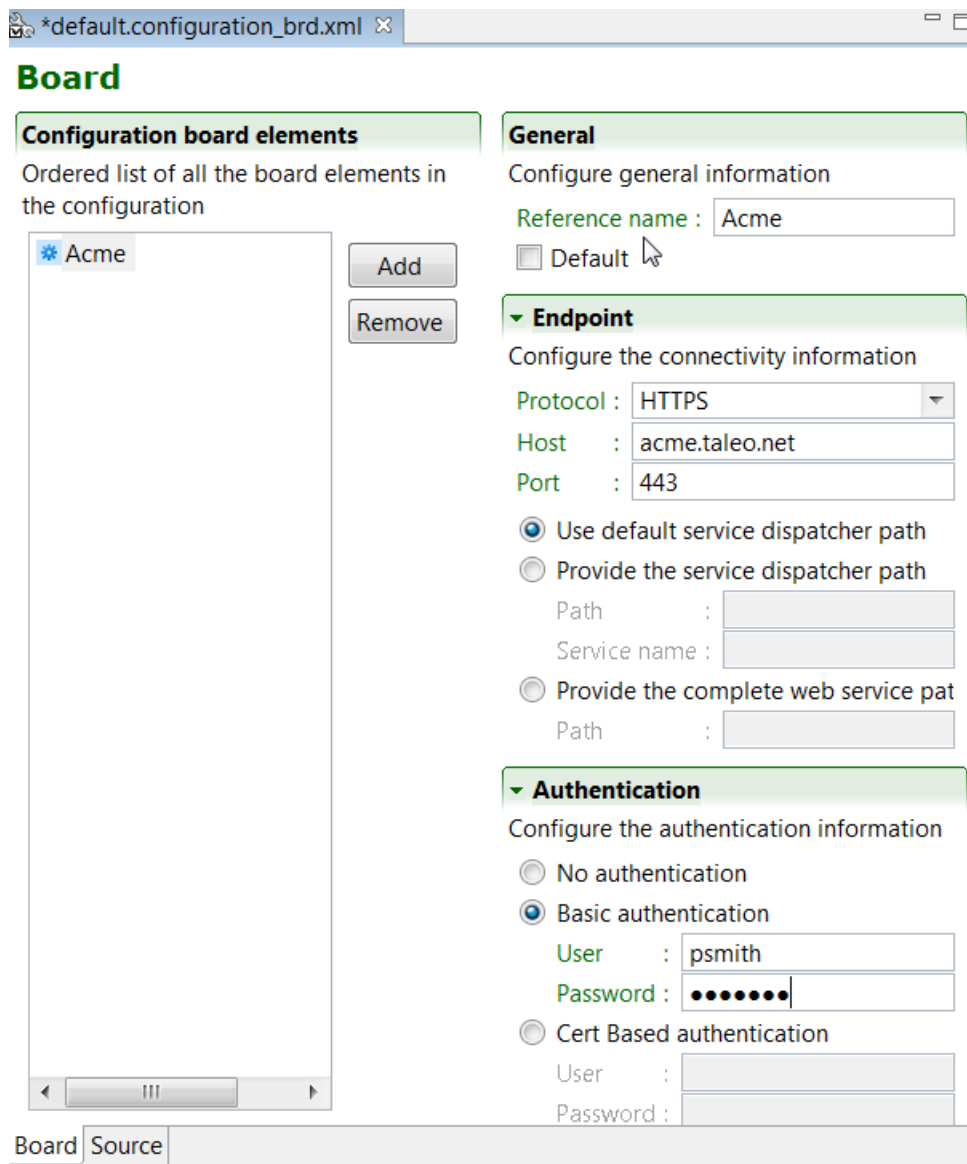
### Board Editor

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 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, 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: No authentication is listed as an option for testing purposes.

Basic Authentication: This is the minimal requirement to connect to the Taleo environment.

Cert Based Authentication: This is used to sign the requests and responses between TCC and Taleo. This is in addition to basic authentication.

For both types of authentication, the user must provide the following information:

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

## Symbol Element

Symbols can be used anywhere in a configuration file and their values are resolved when running 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 a Linux operating system, 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

- **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]("000000000000"))] or [NUMBER\_FORMAT([itk\_attribute\_count]("#,##0.00"))]

- **CFGFOLDER**

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 running TCC using 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
```

- **Open bracket symbol delimiter**

TCC uses the open bracket [ as a symbol delimiter. To use the open bracket more than once it must be doubled.

Normally the regex\_count function is written like this:

```
<quer:customFunction name="REGEXP_COUNT">  
<quer:customFunction name="REGEXP_REPLACE">
```

```
<quer:field path="ProfileInformation,Candidate,HomePhone"/>
<quer:string>[^0-9]+</quer:string>
<quer:string/>
</quer:customFunction>
<quer:string>[:digit:]*</quer:string>
<quer:string>1</quer:string>
<quer:string>i</quer:string>
</quer:customFunction>
```

To work properly in TCC, it must be written like this:

```
<quer:customFunction name="REGEXP_COUNT">
<quer:customFunction name="REGEXP_REPLACE">
<quer:field path="ProfileInformation,Candidate,HomePhone"/>
<quer:string>[^0-9]+</quer:string>
<quer:string/>
</quer:customFunction>
<quer:string>[[[:digit:]]</quer:string>
<quer:string>1</quer:string>
<quer:string>i</quer:string>
</quer:customFunction>
```

## Creating Endpoints

Taleo Connect Client is running.

1. Click 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

Taleo Connect Client is running.

1. Click 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

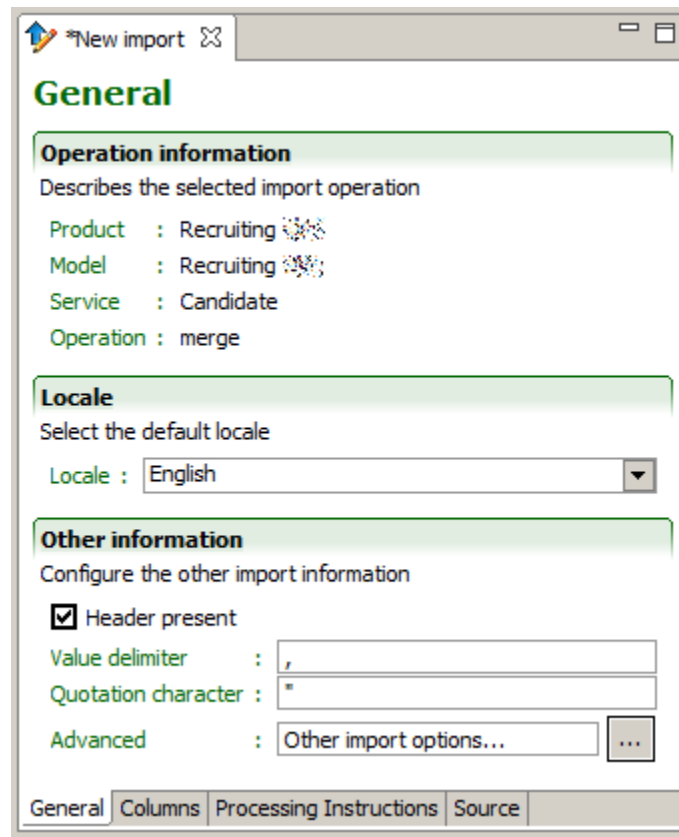
### Import Editor

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.



## 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 runs 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.

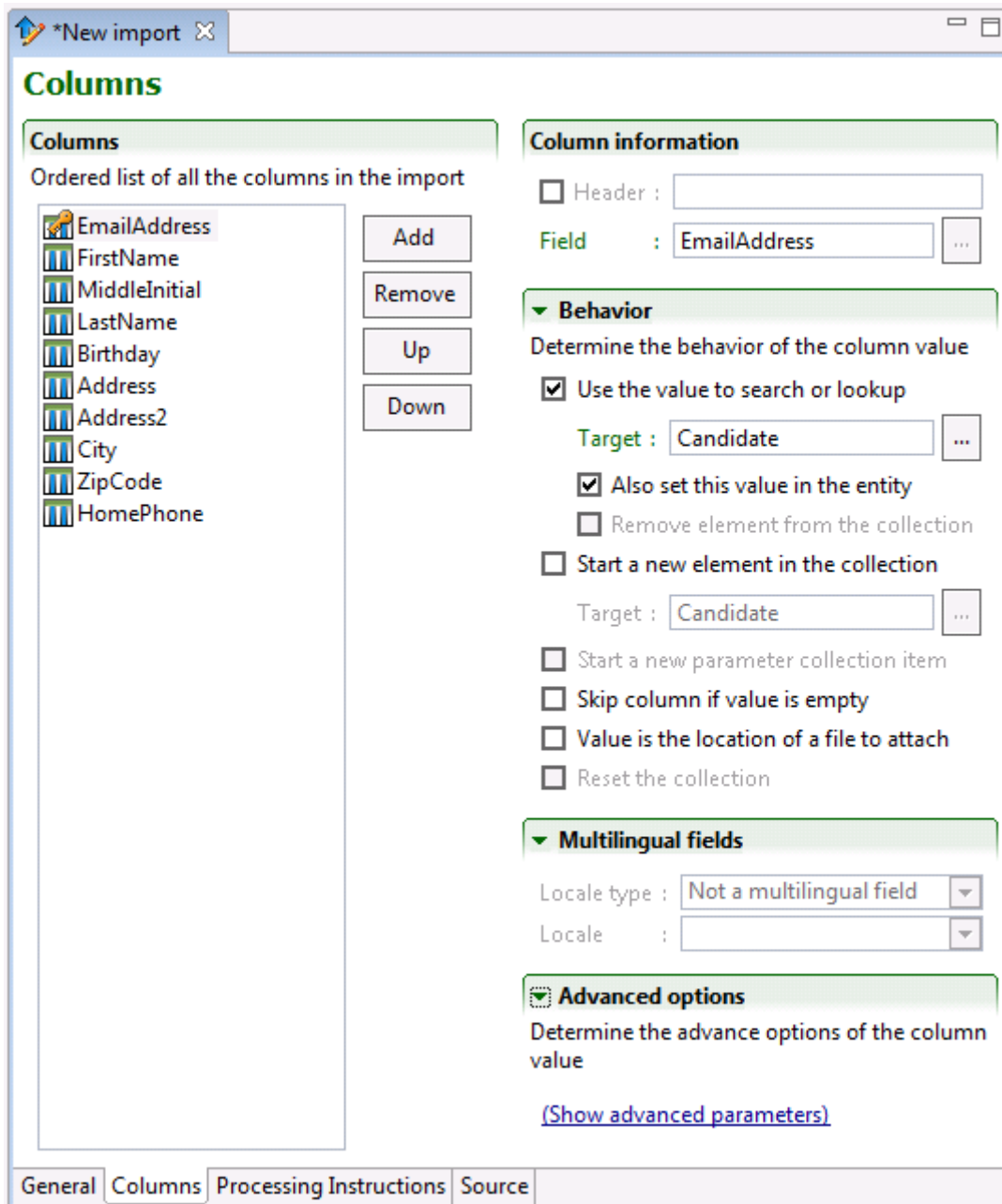
- **switch.system.maintenance**  
This option allows you to automatically switch the zone to maintenance mode when importing Organization, Location, or JobField (OLF), or deleting Department.
- Possible values:
  - always: The integration processor will force the zone to switch into maintenance mode.
  - 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.
- **progress.interval**  
Defines the interval, in number of records processed, used to update the progress of the request in the monitor. For example, setting a value of 100 on a request containing 2000 records, would limit to 20 the number of progress updates sent to TCC.
- Possible values:  
A positive number, the default value is 1.

## 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 multilingual 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 done 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 if value is 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 resettable relations.

## Multilingual Fields

Multilingual 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 done 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.

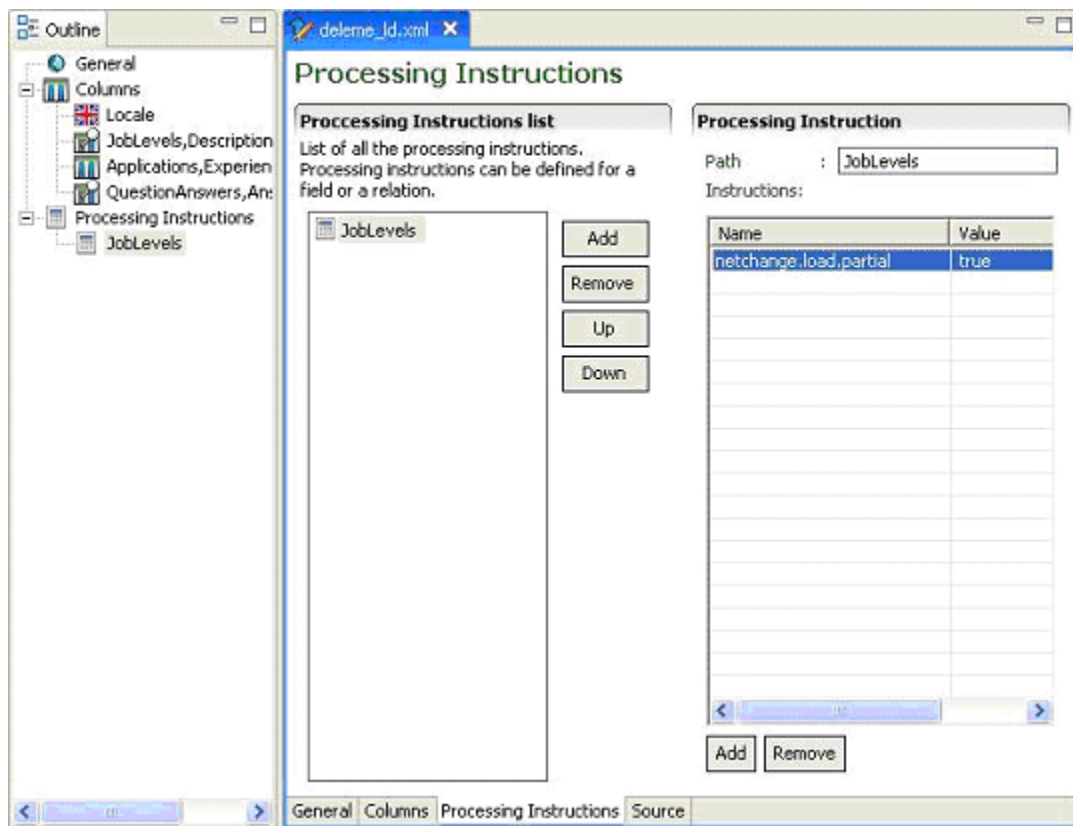
- Possible values:

- 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.

## Creating a New Import File

Make sure the Taleo Connect Client application is open.

1. Click File > New > New Import Wizard >Next.

**New Import Wizard**

This wizard creates a new import specification by selecting a service operation from one of the available product models.

**Select a service**

Product : Recruiting

Model : Recruiting

Filter :

Candidate  
CieEmployer  
CieInstitution  
CieProgram  
CompetencyCategory  
CompetencyGroup  
CorporateRecruitmentSource  
Department  
ExternalApplication  
ExternalCandidate

**Template**

Create a new import specification or use a template

Create new import specification

Operation : merge

Use the following import template

Template : Basic Candidate import

Finish Cancel

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

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

1. Click 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

Make sure the Taleo Connect Client application is open.

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

## Building an Import Specification

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.
4. Save file.

## Creating a CSV File

A configured and saved import specification is open in the import editor.

1. Click the icon 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

1. Open the Preferences window by clicking Window > Preferences > Client > Design.
2. In the Sample CSV viewer, click the Open window button. The Select file window opens.
3. Navigate to the new CSV editor folder.
4. Select the CSV editor.
5. Click Open. The path and name of the new default CSV editor are displayed in the Sample CSV viewer dialog box.
6. Click Apply then OK.

## Export

### Export Editor

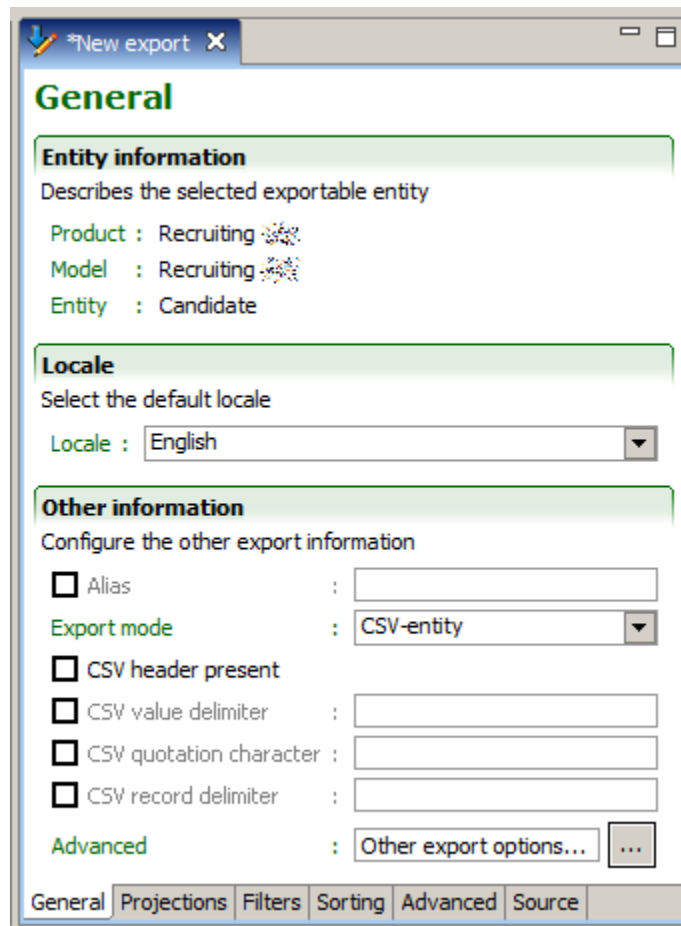
The Export Editor uses a business model from a particular Taleo product version to create a specification for information to be extracted from Taleo products.

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.





## 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.

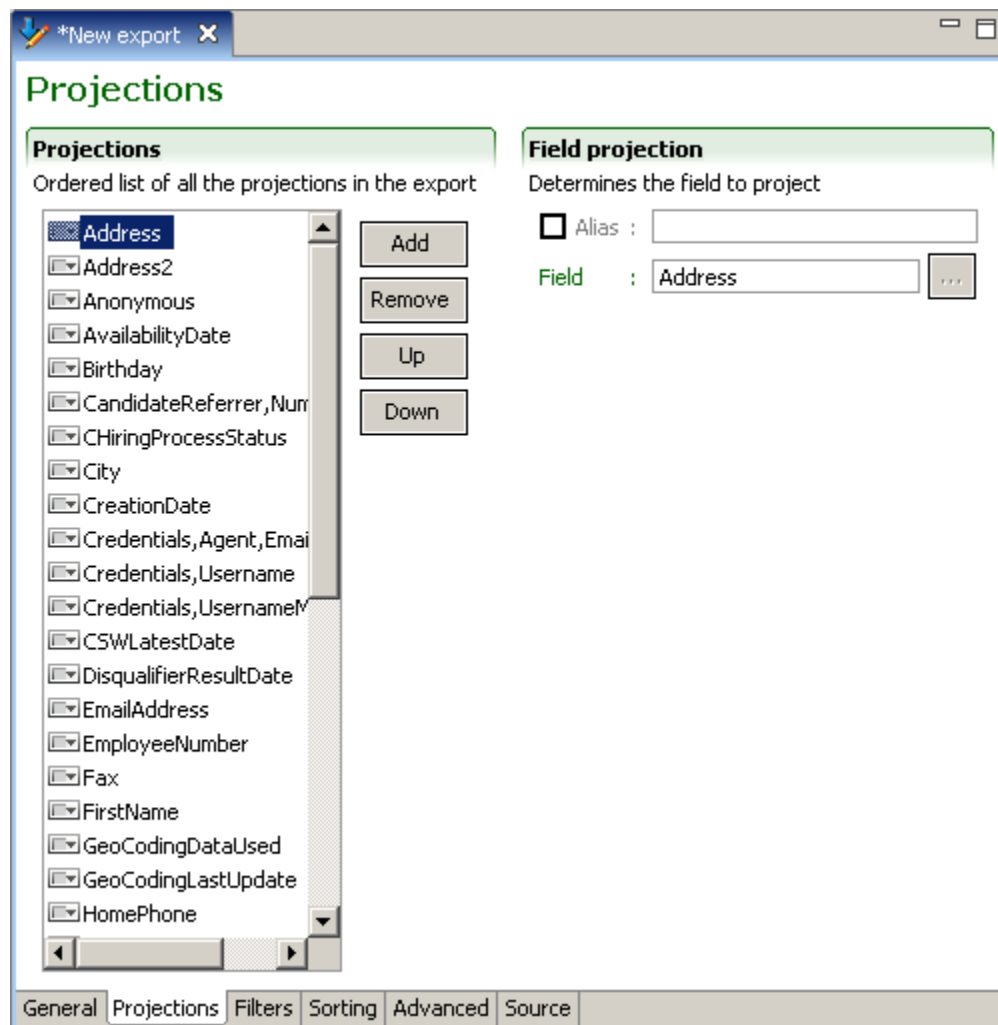
- **false**: the limit and count attributes are absent from the response SOAP document.
- **progress.interval**  
Defines the interval, in number of records processed, used to update the progress of the request in the monitor. For example, setting a value of 100 on a request containing 2000 records, would limit to 20 the number of progress updates sent to TCC.
- Possible values:  
A positive number, the default value is 1.

## 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 multilingual 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**

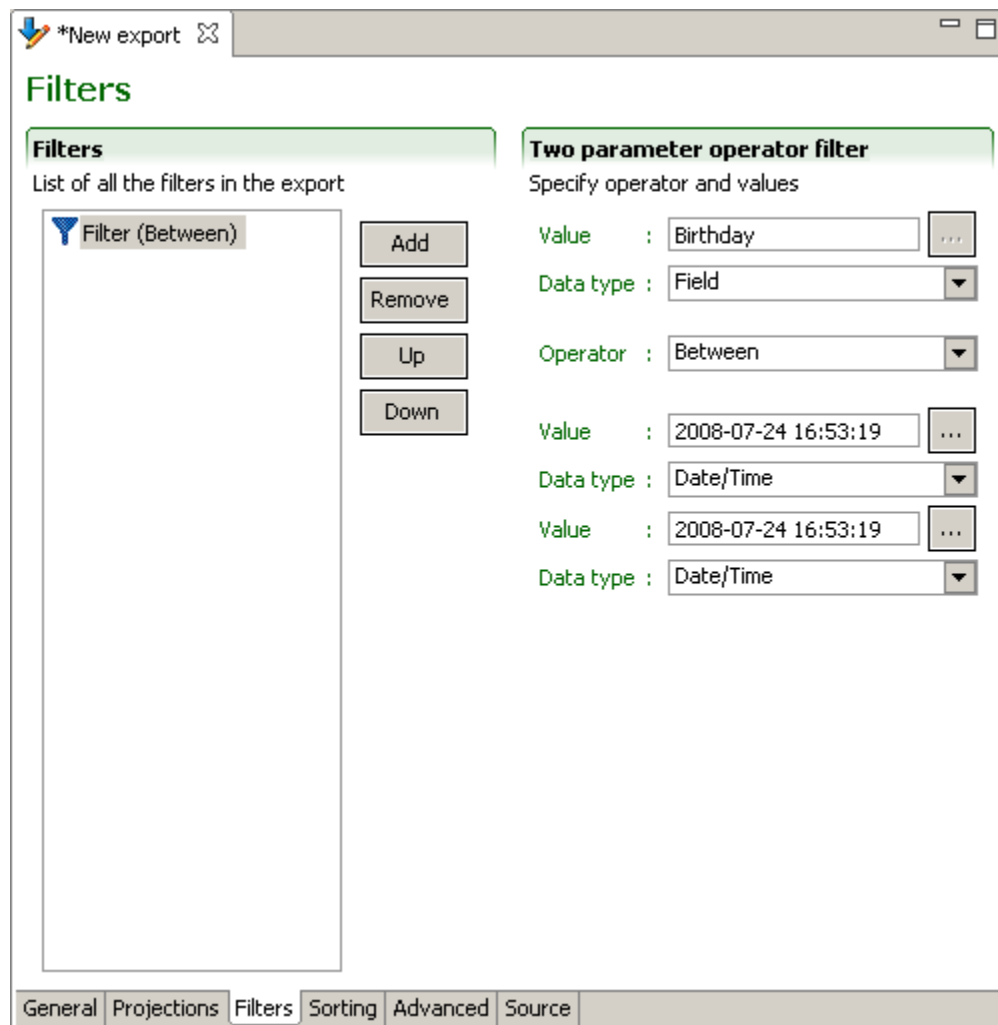
Allows the language code of related entity fields to be exported in CSV-report format. The alias of a multilingual field projection is optional. When an alias is not present, the path itself will be used. The value exported can be the language code (2), or locale code (en).

- **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

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.

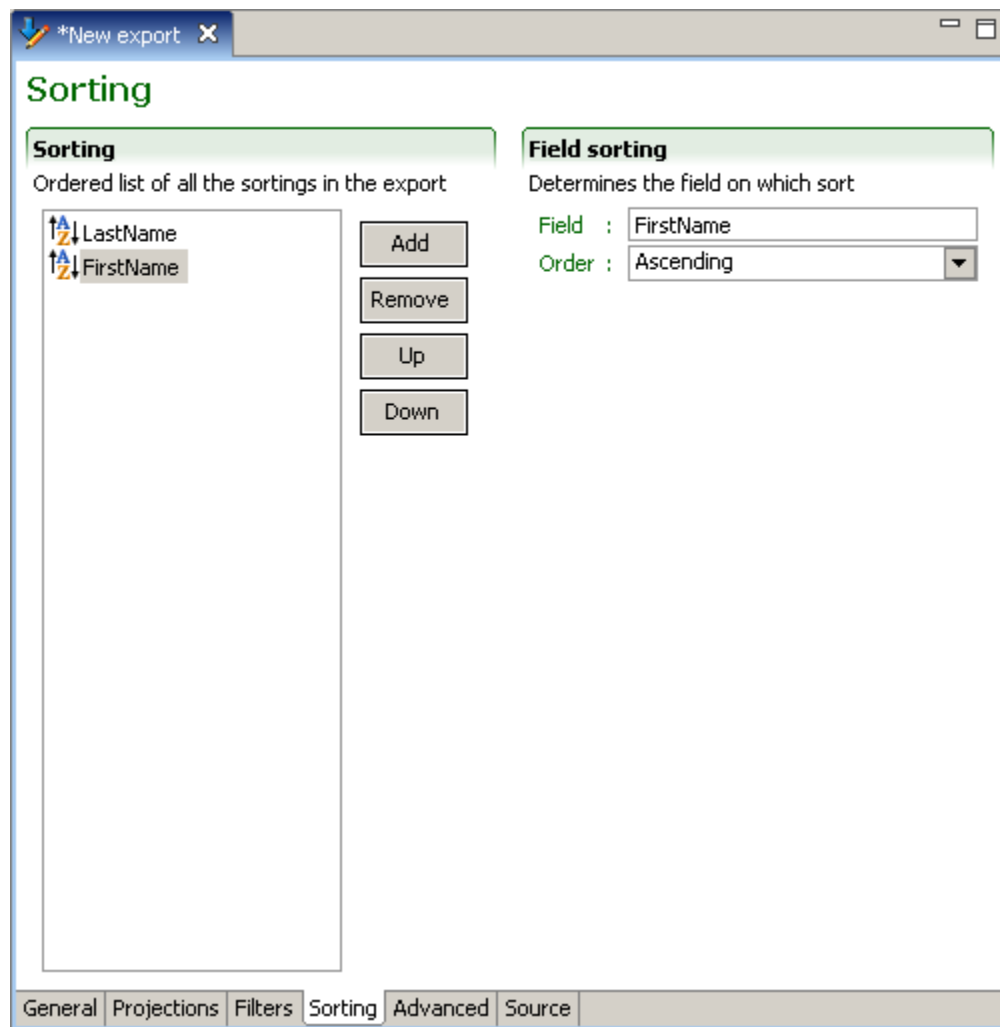
## One/No parameter operator filter

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**



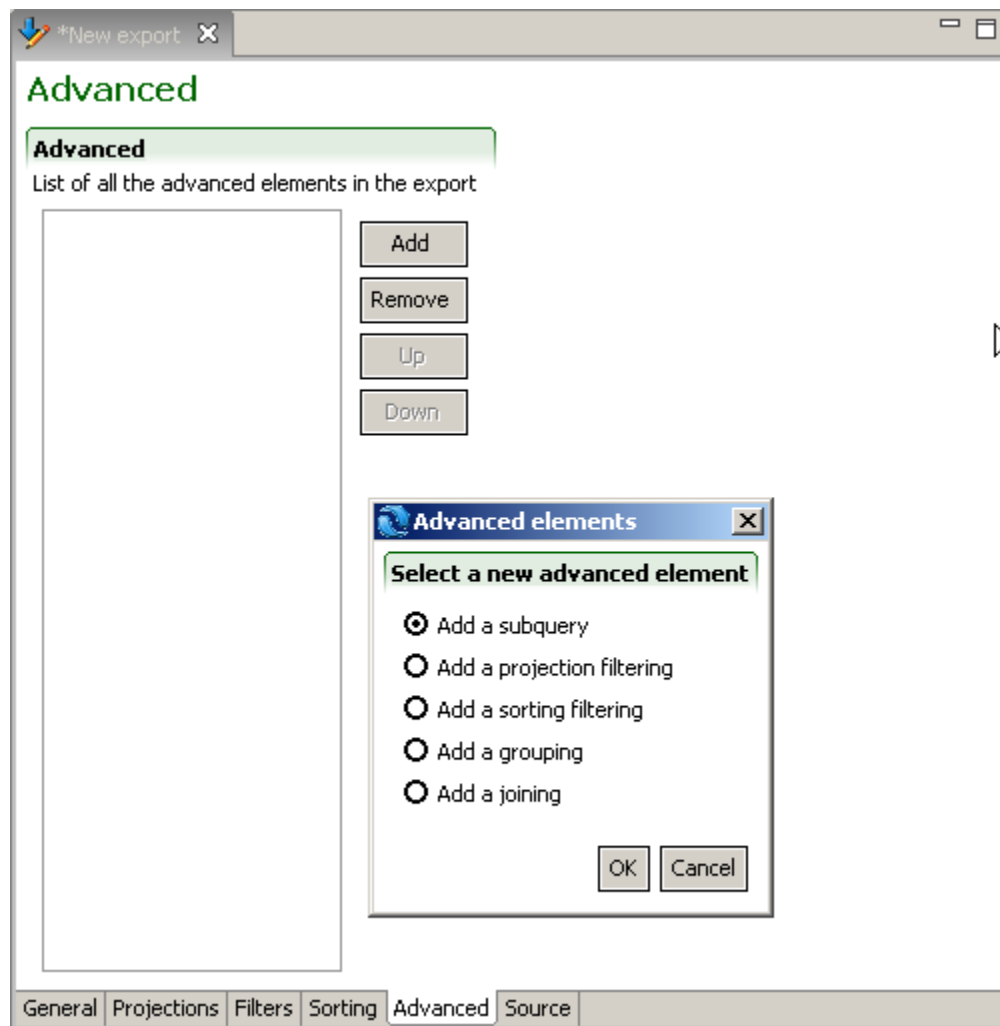
When exporting in T-XML or CSV-entity, field based projection sorting allows specifying the sorting order of sub-entities. The only option is to select the Order of the sort, which can either be Ascending or Descending.

- **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

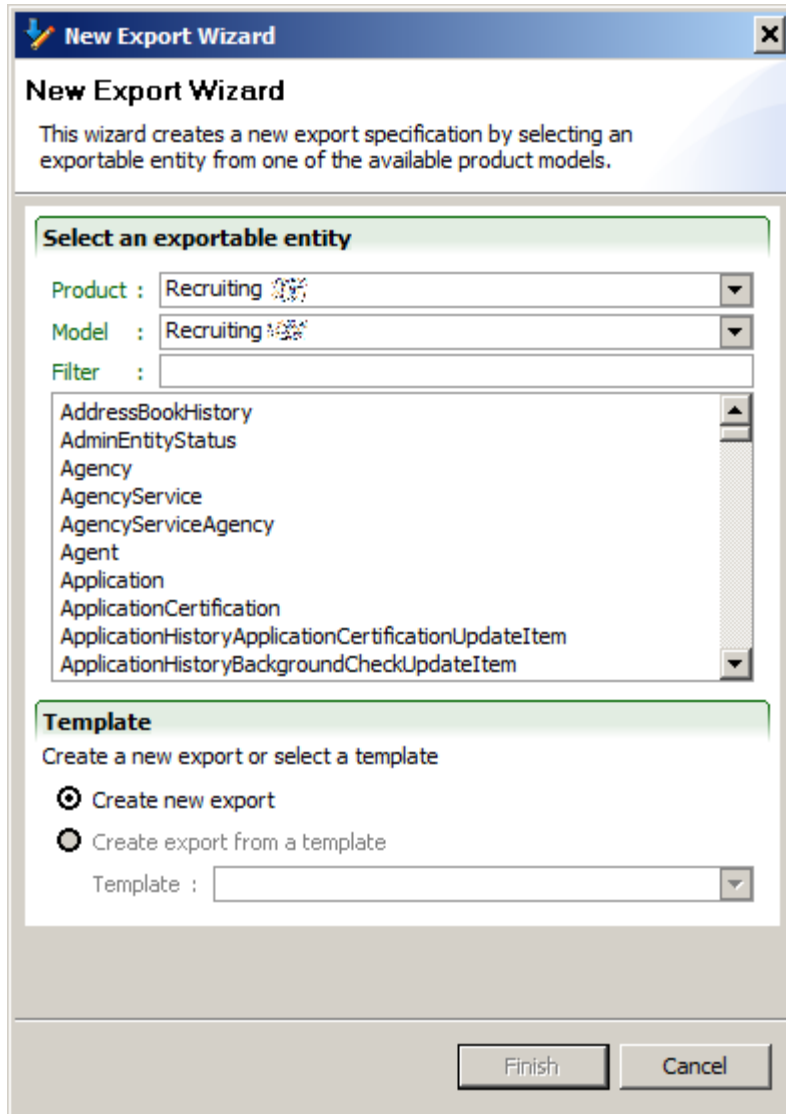
The remove button deletes selected elements. The Add button creates new elements:

- subquery
- projection filtering
- sorting filtering
- grouping
- joining

## Creating a New Export File

Make sure the Taleo Connect Client application is open.

1. Click 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

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

1. Click 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

Make sure the Taleo Connect Client application is open.

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

## Building an Export Specification

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.

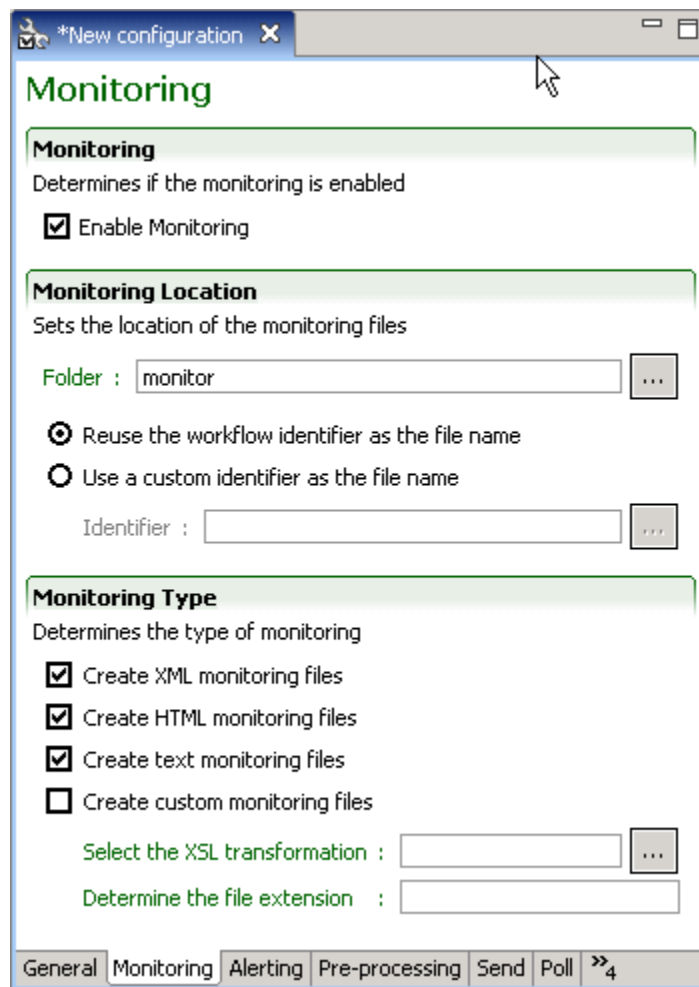
The Encoding states the way the file is encoded. The Taleo environment only supports UTF-8 encoding. Conversion is available in the Pre- and post-processing steps.

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. The **Reuse the workflow identifier as the file name** setting will use the workflow identifier as defined in **General Tab > Workflow > Workflow identifier** for the monitoring file name. The **Use a custom identifier** setting allows you to specify the monitoring file name instead of using reusing the workflow identifier as the file name. In addition, the monitoring file name can use either a static or a unique identifier:

- Use a static identifier. A static or constant identifier for the monitoring file name gives you easy access to the result of a given integration process over time. Each workflow execution would overwrite the previous one, but a simple shortcut could be added to an Internet browser to quickly access the current result.



By default, the monitoring file name is static.

- Use a unique identifier. A unique identifier creates a different monitoring file name for each workflow execution. This ensures that the monitoring information is always available and easily retrievable for a particular execution.

A unique identifier can be used either when the file name reuses the workflow identifier or when the file name uses a custom identifier.

There are two ways to add a unique identifier to a monitoring file name:

- a. Append a token (UUID, time stamp) to the file name.
- b. Add a system property named "com.taleo.integration.client.uniquemonitoringid=true" to the command line or set "-Dcom.taleo.integration.client.uniquemonitoringid=true" in the ini file.

## 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.

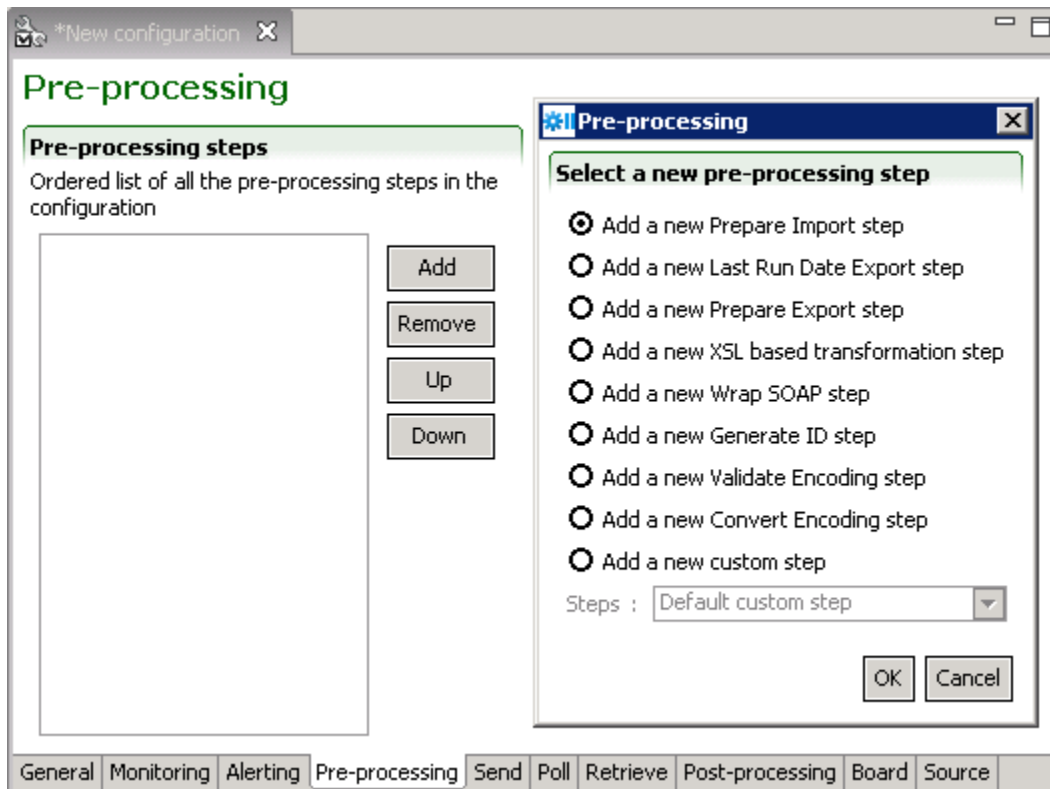
Subject determines the subject line of the alert e-mail.

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 perform 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 Setup**

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.

**Attachment Maximum File Size:** The maximum file size is now defaulted to 100MB (104857600 bytes). This can be modified by setting a different value, in bytes, for the `attachment.max.file.size` advanced parameter. For example, `attachment.max.file.size=104857600`

**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.

This applies only to Net Change steps, in all other cases the notification email is sent no matter what choice is selected.

## 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.

- No authentication: No authentication is listed as an option for testing purposes.
- Basic Authentication: This is the minimal requirement to connect to the Taleo environment.

- **Cert Based Authentication:** This is used to sign the requests and responses between TCC and Taleo. This is in addition to basic authentication.

For both types of authentication, the user must provide the following information:

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

## 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.

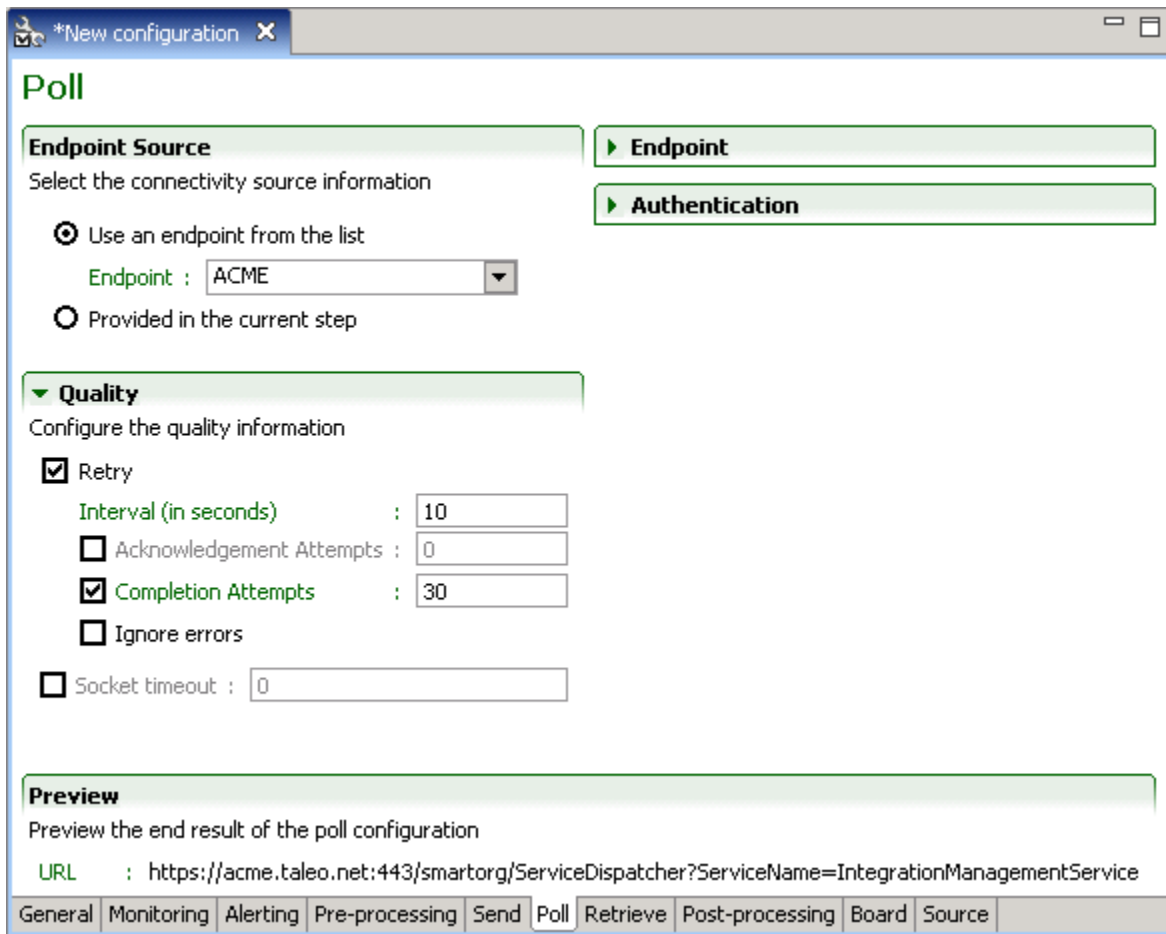
Validate request is a legacy option only offered in RS-XML workflows where the request message was pre-validated.

## 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.

- No authentication: No authentication is listed as an option for testing purposes.
- Basic Authentication: This is the minimal requirement to connect to the Taleo environment.
- Cert Based Authentication: This is used to sign the requests and responses between TCC and Taleo. This is in addition to basic authentication.

For both types of authentication, the user must provide the following information:

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

## 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. This setting only applies to polling errors and not document errors.

The workflow will stop when the following document states are encountered: Interrupted, Suspended, 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.

## 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.

## 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.

- No authentication: No authentication is listed as an option for testing purposes.
- Basic Authentication: This is the minimal requirement to connect to the Taleo environment.

- **Cert Based Authentication:** This is used to sign the requests and responses between TCC and Taleo. This is in addition to basic authentication.

For both types of authentication, the user must provide the following information:

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

## 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.

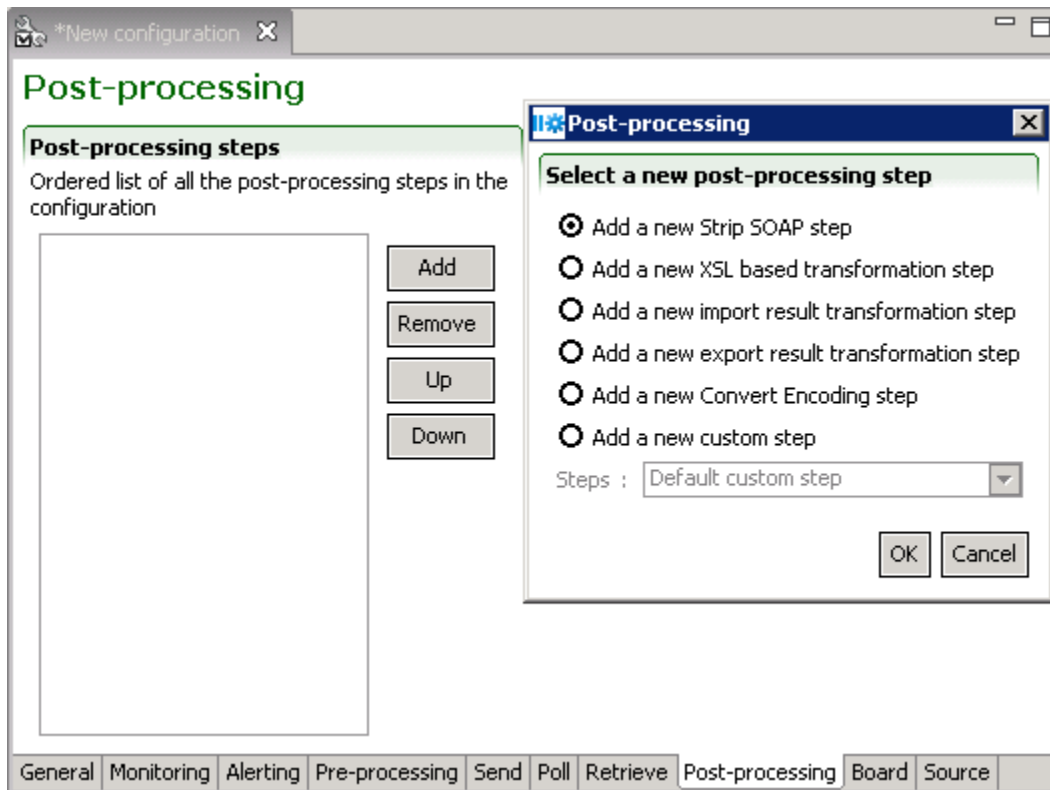
Validate request is a legacy option only offered in RS-XML workflows where the request message was pre-validated.

## 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 perform 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.

### 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.

**Attachment Maximum File Size:** The maximum file size is now defaulted to 100MB (104857600 bytes). This can be modified by setting a different value, in bytes, for the `attachment.max.file.size` advanced parameter. For example, `attachment.max.file.size=104857600`

**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.

This applies only to Net Change steps, in all other cases the notification email is sent no matter what choice is selected.

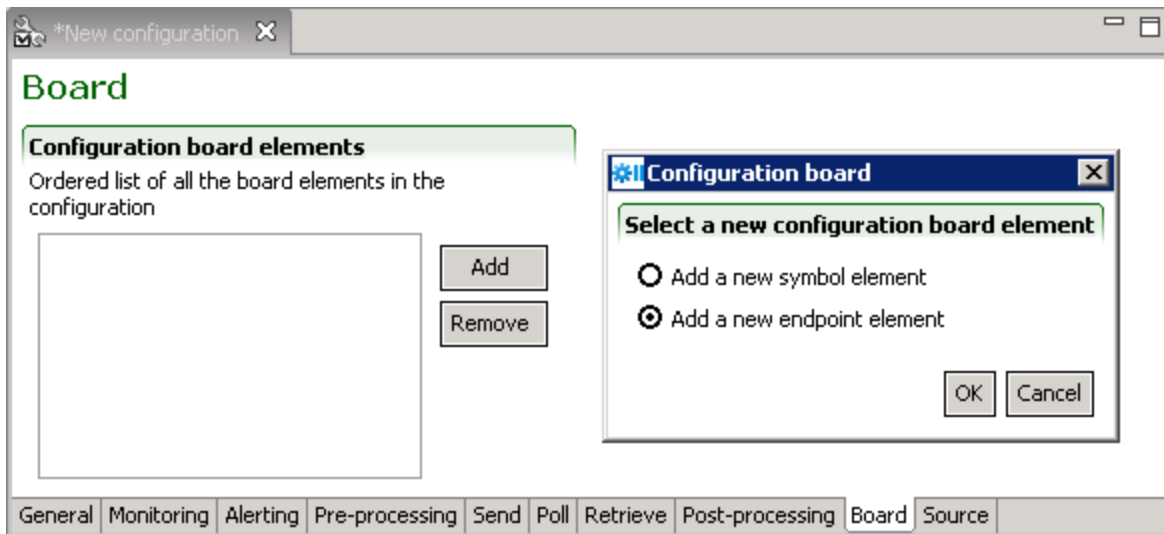
- **Report Filtering**

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.



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

Make sure the Taleo Connect Client application is open.

1. Click File > New > New Configuration Wizard > Next.
2. Select the configuration file base.
3. Click the Navigate icon.
4. Select directory.
5. Select file.
6. Click Open.
7. Click Next.
8. Verify connectivity information.
9. Click Finish.

The New configuration specification is displayed in the configuration editor.

## Saving a New Configuration File

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

1. Click 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

Make sure the Taleo Connect Client application is open.

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

## Executing a Configuration

Make sure a configuration file is open and active.

1. Click Execute the configuration icon.

## Runtime

### 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

A workflow configuration file is available for execution.

1. Click Window > Open Perspective > Runtime. The Runtime Perspective opens.
2. Click File > Open Configuration.
3. Select directory.
4. Select file.
5. Click Open. The configuration file opens.
- 6.
7. Click the Execute the configuration icon.
8. Maximise the Runtime Perspective.

The workflow execution details are displayed.

## Console

### Console Perspective

The Console perspective displays a real-time view of all integration processes 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 message 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

1. Click File > New > New Profile Wizard > Next.
2. Select the profile type.
  - o **Endpoint:** Create the profile based on an endpoint. Select product from pull down menu. Select Endpoint from pull down menu
  - o **Blank profile:** Create a blank profile and fill the connectivity information manually.
  - o **Demo profile:** Use 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

Make sure a new console profile is open in the message viewer.

1. Click 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 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>\_<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, Linux).

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

## Scheduling in Windows

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

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:

```
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

-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 occurs, 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 instruction ensures that the proper version of the Xerces XML parser is used. 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.integra-tion.client.Client.execute()`. The JavaDoc provided for this method in the `docs/api` folder of the product describes the behavior and parameter values.





# 4 Getting Started

## 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@abc.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 2011  
Analyst at XYZ Inc. for the first half of 2012

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

### Creating Starter Folder

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.

**4. File>New>Folder.**

- 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.

**6. Minimize window.**

## Creating Curriculum Vitae

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

**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.

### 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

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 FileName from the Entity Structure to the Columns list.
- c. Drag and drop FileContent from the Entity Structure to the Columns list.

### 8. Click AttachedFiles,FileContent 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.

### 9. Scroll Entity Structure to TalentUser and expand.

In terms of the business model, the TalentUser relation is where we create a general profile for the candidate.

10. Scroll Entity Structure to Profile and expand.
11. Drag and drop Locale from the Entity Structure to the Columns list.  
The language of the profile is a mandatory field.
  - a. Click TalentUser,Profile,Locale in Columns list.
  - b. Select Header check box in Column information section.
  - c. Enter Locale as header label.
12. Scroll Profile to Experiences and expand.
13. Drag and drop DisplaySequence from the Entity Structure to the Columns list.  
The display sequence of the experience is a mandatory field.
14. Click TalentUser,Profile,Experiences,DisplaySequence in Columns list.
  - a. Select Header check box in Column information section.
  - b. Enter Exp1-Sequence as header label.
15. Drag and drop OtherEmployerName from the Entity Structure to the Columns list.
16. Click TalentUser,Profile,Experiences,OtherEmployerName in Columns list.
  - a. Select Header check box in Column information section.
  - b. Enter Exp1-Employer as header label.
17. Drag and drop OtherJobTitle from the Entity Structure to the Columns list.
18. Click TalentUser,Profile,Experiences,OtherJobTitle in Columns list.
  - a. Select Header check box in Column information section.
  - b. Enter Exp1-JobTitle as header label.
19. Drag and drop BeginDate from the Entity Structure to the Columns list.
20. Click TalentUser,Profile,Experiences,BeginDate in Columns list.
  - a. Select Header check box in Column information section.
  - b. Enter Exp1-BeginDate as header label.
21. Drag and drop EndDate from the Entity Structure to the Columns list.
22. Click TalentUser,Profile,Experiences,EndDate in Columns list.
  - a. Select Header check box in Column information section.
  - b. Enter Exp1-EndDate as header label.
23. 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
24. Click TalentUser,Profile,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.
25. Drag and drop OtherEmployerName from the Entity Structure to the Columns list.
26. Click TalentUser,Profile,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.
27. Drag and drop OtherJobTitle from the Entity Structure to the Columns list.
28. Click TalentUser,Profile,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.
29. Drag and drop BeginDate from the Entity Structure to the Columns list.
30. Click TalentUser,Profile,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.
31. Drag and drop EndDate from the Entity Structure to the Columns list.
32. Click TalentUser,Profile,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

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

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@abc.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.

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

The New Configuration Wizard window opens.

2. Select Based on an import specification.
  - a. Click the Browse icon.
  - 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

1. Verify general tab information and correct if necessary.
2. In Request select Pre-defined value.
  - a. In file, click the Browse icon.
  - 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 the Browse icon.
  - b. Select C:\tcc\starter.
  - c. In Identifier, click the Browse icon.

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 the Browse icon.
  - 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

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

Import configuration file candidate-merge\_cfg.xml is open and selected.

1. **Window>Open perspective>Runtime**
2. Click Execute the configuration icon.
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.

## Exporting

### Creating Export Specification

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

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 icon.
- 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@abc.com in second Value.
- g. Verify Second Data type is String.



## Saving Export Specification

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

1. **File>New>New Configuration Wizard**  
The New Configuration Wizard window opens.
2. Select Based on an export specification.
  - a. Click the Browse icon.
  - 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

1. Verify general tab information and correct if necessary.
2. In Request select Pre-defined value.
  - a. In file, click the Browse icon.
  - 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 the Browse icon.
  - b. Select C:\tcc\starter.
  - c. In Identifier, click the Browse icon.  
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 the Browse icon.
        - 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

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

Export configuration file candidate\_export\_cfg.xml is open and selected.

1. **Window>Open perspective>Runtime**
2. Click the Execute the configuration icon.
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.

# 5 Using TCC

## Hierarchical Entities

### 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.

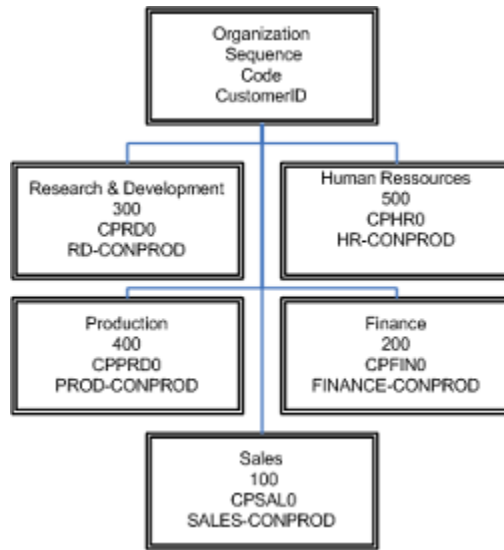
Rigid hierarchical entities require special attention in import operations. To create a rigid type entity you must specify its Parent. To delete a rigid type entity you must first delete all its children or move them to a new Parent.

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.

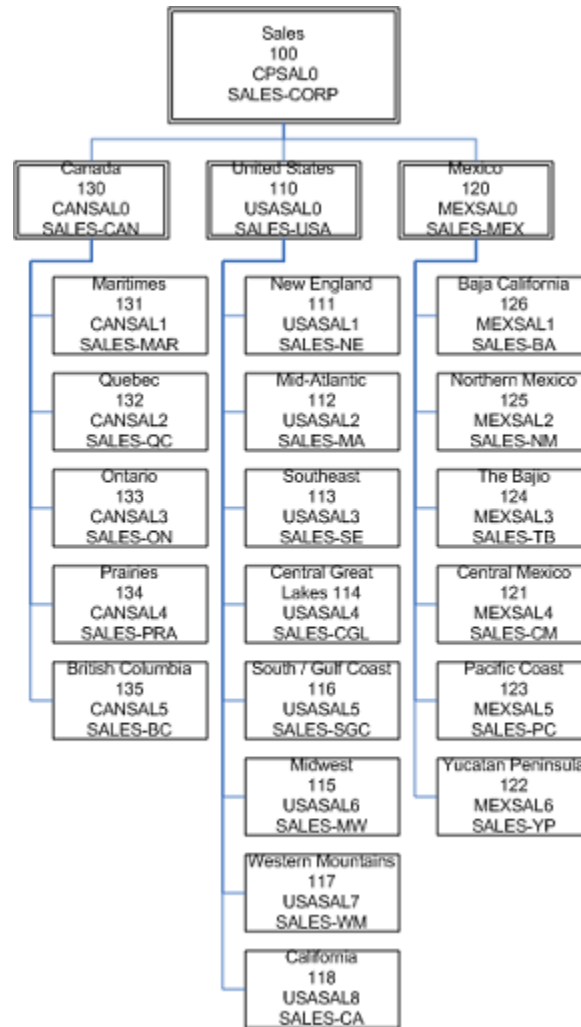
#### Model

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.

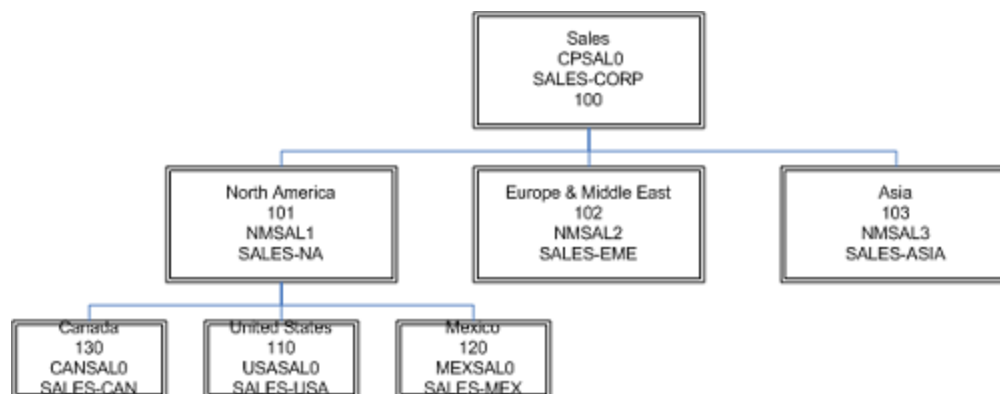
Here is the top of the organization:



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.

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

1. Click File, New to open the New Export Wizard.
2. In Product pull down menu, Select SmartOrg [Version].
  - a. In Filter, enter letter o.
  - b. Click Organization.
  - c. Select Create new export.
  - d. Click 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.
5. Drag and Drop Code, Name, and Level from Entity view to Projections list.
6. 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.
7. Save as OrgRoot-export\_sq.xml in organization-export folder. Level 0 is the root level.
8. Click File, New to open the New Configuration Wizard.
9. Select Based on an export specification.
  - a. Click the Browse icon.
  - 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.
10. In General tab, under Request, select Pre-defined value.
  - a. Click the Browse icon. A Select file window opens.

- b. Navigate to organization-export folder.
  - c. Select OrgRoot-export\_sq.xml.
  - d. Click Open. The file path and name are displayed in the dialog box.
11. In General tab, under Response, select Generated using an identifier.
  - a. Click the Browse icon.
  - b. Navigate to organization-export folder.
  - c. Click OK. The file path is displayed in the dialog box.
  - d. Click the Browse icon.
  - e. In the Template dialog box enter [FILE]-[NOW].
  - f. Click OK. [FILE]-[NOW] is displayed in the dialog box.
12. Save as OrgRoot-Extract\_cfg.xml in organization-export folder.
13. Click the Execute the configuration icon. and wait for the workflow to complete the extraction.
14. Open organization-export folder.
15. Open CSV result file OrgRoot-export\_sq-yyyyMMddTHHmss.csv in a csv editor or notepad

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

```
Code, Name, Level
ROOT-ORGANIZATION, [en]=New Millenium, 0
```

## Creating an Organization

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

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 the Browse icon.
  - 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\_Id.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.

```
Code, CustomerID, Name, ParentCode, IndustryNumber, DisplaySequence
CPCORP0, CORP-CONPROD, Corporate Headquarters, ROOT-CODE, 12, 1
CPRD0, RD-CONPROD, Research & Development, CPCORP0, 12, 300
CPHR0, HR-CONPROD, Human Ressources, CPCORP0, 12, 500
CPPRD0, PROD-CONPROD, Production, CPCORP0, 12, 400
CPFIN0, FINANCE-CONPROD, Finance, CPCORP0, 12, 200
CPSAL0, SALES-CONPROD, Sales, CPCORP0, 12, 100
CANSAL0, SALES-CAN, Canada, CPSAL0, 12, 130
USASAL0, SALES-USA, United States of America, CPSAL0, 12, 110
MEXSAL0, SALES-MEX, Mexico, CPSAL0, 12, 120
CANSAL1, SALES-MAR, Maritimes, CANSAL0, 12, 131
CANSAL2, SALES-QC, Quebec, CANSAL0, 12, 132
CANSAL3, SALES-ON, Ontario, CANSAL0, 12, 133
CANSAL4, SALES-PRA, Prairies, CANSAL0, 12, 134
CANSAL5, SALES-BC, British Columbia, CANSAL0, 12, 135
USASAL1, SALES-NE, New England, USASAL0, 12, 111
USASAL2, SALES-MA, Mid-Atlantic, USASAL0, 12, 112
USASAL3, SALES-SE, Southeast, USASAL0, 12, 113
USASAL4, SALES-CGL, Central Great Lakes, USASAL0, 12, 114
USASAL5, SALES-SGC, South / Gulf Coast, USASAL0, 12, 116
USASAL6, SALES-MW, Midwest, USASAL0, 12, 115
USASAL7, SALES-WM, Western Mountains, USASAL0, 12, 117
USASAL8, SALES-CA, California, USASAL0, 12, 118
MEXSAL1, SALES-BA, Baja California, MEXSAL0, 12, 126
MEXSAL2, SALES-NM, Northern Mexico, MEXSAL0, 12, 125
MEXSAL3, SALES-TB, The Bajio, MEXSAL0, 12, 124
MEXSAL4, SALES-CM, Central Mexico, MEXSAL0, 12, 121
MEXSAL5, SALES-PC, Pacific Coast, MEXSAL0, 12, 123
MEXSAL6, SALES-YP, Yucatan Peninsula, MEXSAL0, 12, 122
```

**10.** Save file as organization-create.csv in organization-create folder.**11.** From TCC open New Configuration Wizard **File>New>New Configuration Wizard**.

The New Configuration Wizard opens.

**12.** Select Based on an import specification.

- a. Click the Browse icon.  
A Select folder window opens.
- b. Navigate to organization-create folder.
- c. Select organization-create\_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 the Browse icon.  
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 the Browse icon.

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 the Browse icon.

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 the Execute the configuration icon and wait for the workflow to complete the import.
- 17. Open organization-create folder.
- 18. Open CSV result file organization-create-yyyyMMddTHHmss.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 multilingual fields with the specific locale method.

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

Taleo application has more than one active language.

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 the 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 Organisation: name.
  - 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 Multilingual 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 Multilingual fields, Locale type select Specific locale from pull down menu.
    - d. In Multilingual 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 Multilingual fields, Locale type select Specific locale from pull down menu.
    - f. In Multilingual fields, Locale select spanish from pull down menu.
9. Save as organization-update\_id.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,Investigatiión y desarrollo,300
CPHR0,HR-CONPROD,Human Ressources,Ressources humaines,Recursos humanas,500
CPPRD0,PROD-CONPROD,Production,Production,Producción,400
CPFIN0,FINANCE-CONPROD,Finance,Finance,Finanzas,200
CPSAL0,SALES-CONPROD,Sales,Ventes,Ventas,100
CANSAL0,SALES-CAN,Canada,Canada,Canada,130
USASAL0,SALES-USA,United States of America,États Unis d'Amérique,Estados Unidos de América,110
```

MEXSAL0, SALES-MEX, Mexico, Mexique, Mexico, 120  
CANSAL1, SALES-MAR, Maritimes, Maritimes, Marítimos, 131  
CANSAL2, SALES-QC, Quebec, Québec, Quebec, 132  
CANSAL3, SALES-ON, Ontario, Ontario, Ontario, 133  
CANSAL4, SALES-PRA, Prairies, Prairies, Praderas, 134  
CANSAL5, SALES-BC, British Columbia, Colombie Britannique, Colombia Británica, 135  
USASAL1, SALES-NE, New England, Nouvelle Angleterre, Nueva Inglaterra, 111  
USASAL2, SALES-MA, Mid-Atlantic, Atlantique centre, Atlántico central, 112  
USASAL3, SALES-SE, Southeast, Sud-est, Sudeste, 113  
USASAL4, SALES-CGL, Central Great Lakes, Grands lacs central, Gran lagos central, 114  
USASAL5, SALES-SGC, South / Gulf Coast, Sud / Côte du golfe, Sur / Costa del golfo, 116  
USASAL6, SALES-MW, Midwest, Midwest, Midwest, 115  
USASAL7, SALES-WM, Western Mountains, Montagnes de l'ouest, Montañas del oeste, 117  
USASAL8, SALES-CA, California, Californie, California, 118  
MEXSAL1, SALES-BA, Baja California, Baja California, Baja California, 125  
MEXSAL2, SALES-NM, Northern Mexico, Mexique du Nord, Mexico del norte, 125  
MEXSAL3, SALES-TB, The Bajio, Le Bajio, El Bajío, 124  
MEXSAL4, SALES-CM, Central Mexico, Mexique central, Mexico central, 121  
MEXSAL5, SALES-PC, Pacific Coast, Côte du Pacifique, Costa del Pacífico, 123  
MEXSAL6, SALES-YP, Yucatan Peninsula, Péninsule du Yucatan, Península del Yucatan, 122

12. Save file as organization-update.csv in organization-update folder.
13. From TCC open New Configuration Wizard **File>New>New Configuration Wizard**.

The New Configuration Wizard opens.

14. Select Based on an import specification.
  - a. Click the Browse icon.  
A Select folder window opens.
  - b. Navigate to organization-update folder.
  - c. Select organization-update\_ld.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 the Browse icon.  
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.

16. In General tab, under Response, select Generated using an identifier.
  - a. Click the Browse icon.  
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 the Browse icon.

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 the Execute the configuration icon and wait for the workflow to complete the import.
19. Open organization-update folder.
20. Open CSV result file organization-update-yyyyMMddTHHmss.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

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

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.

```
Code, CustomerID, Name, ParentCode, IndustryNumber, DisplaySequence
NMSAL1, SALES-NA, North America, CPSAL0, 12, 101
NMSAL2, SALES-EME, Europe & Middle East, CPSAL0, 12, 102
NMSAL3, SALES-ASIA, Asia, CPSAL0, 12, 103
```

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 the Browse icon.

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 the Execute the configuration icon and wait for the workflow to complete the import.
10. Open organization-create folder.
11. Open CSV result file reorganization-create-yyyyMMddTHHmss.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

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

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 the Browse icon.
  - 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\_ld.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.
11. From TCC open New Configuration Wizard **File>New>New Configuration Wizard**.

The New Configuration Wizard opens.

12. Select Based on an import specification.
  - a. Click the Browse icon.  
A Select folder window opens.
  - b. Navigate to organization-move folder.
  - c. Select organization-move\_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 the Browse icon.  
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 the Browse icon.  
A Select folder window opens.
  - b. Navigate to organization-move folder.
  - c. Click on OK.  
The file path is displayed in the dialog box
  - d. Click the Browse icon.  
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 the Execute the configuration icon and wait for the workflow to complete the import.
17. Open organization-move folder.
18. Open CSV result file organization-move-yyyyMMddTHHmms.csv in a csv editor or notepad

The following results are displayed.

```
"Index", "Identifier", "Status", "TransactionType", "Result", "Message"
"1", "", "success", "organization.move", , ""
"2", "", "success", "organization.move", , ""
```

```
"3","","success","organization.move",,""
```

## Deleting an Organization

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

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 the Browse icon.
  - 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 delete operation offers two parameters, the Organization to delete and the Reassignorganization 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 ReassignOrganisation and click Remove.
8. Save as organization-delet\_id.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.
12. From TCC open New Configuration Wizard **File>New>New Configuration Wizard**.

The New Configuration Wizard opens.

13. Select Based on an import specification.
  - a. Click the Browse icon.  
A Select folder window opens.
  - b. Navigate to organization-delete folder.
  - c. Select organization-delete\_id.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 the Browse icon.  
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 the Browse icon.  
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 the Browse icon.  
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 the Execute the configuration icon and wait for the workflow to complete the import.
18. Open organization-delete folder.
19. Open CSV result file organization-delete-yyyyMMddTHHmms.csv in a csv editor or notepad

The following results are displayed.

```
"Index","Identifier","Status","TransactionType","Result","Message"
"1","","success","organization.delete",,""
"2","","success","organization.delete",,""
```



```
"3", "", "success", "organization.delete", , ""  
"4", "", "success", "organization.delete", , ""  
"5", "", "success", "organization.delete", , ""  
"6", "", "success", "organization.delete", , ""  
"7", "", "success", "organization.delete", , ""
```

## 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.

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

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.
  - o TEE 7.0.5 and higher : Use date from the zone (default).
  - o 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

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.

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.

**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 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. Select UserAccount,Groups,Description in the column list.
- a. In Behavior section, select Use the value for search or lookup.
- b. 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.

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

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 CPCORPO.
    - 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.
12. From TCC open New Configuration Wizard **File>New>New Configuration Wizard**.  
The New Configuration Wizard opens.
13. Select Based on an export specification.
  - a. Click the 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 the 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 the 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 the 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 the Execute the configuration icon and wait for the workflow to complete the extraction.
18. Open organization-export folder.
19. Open CSV result file OrgLocale-yyyyMMddTHHmss.csv in a csv editor or notepad

```
ParentName,Parentcode,ChildName,ChildLocale  
Corporate Headquarters,CPCORP0,Research & Development,en  
Corporate Headquarters,CPCORP0,Investigatiión y desarrollo,es  
Corporate Headquarters,CPCORP0,Recherche et développement,fr  
Corporate Headquarters,CPCORP0,Human Ressources,en
```

```
Corporate Headquarters,CPCORP0,Recursos humanas,es
Corporate Headquarters,CPCORP0,Ressources humaines,fr
Corporate Headquarters,CPCORP0,Production,en
Corporate Headquarters,CPCORP0,Producción,es
Corporate Headquarters,CPCORP0,Production,fr
Corporate Headquarters,CPCORP0,Finance,en
Corporate Headquarters,CPCORP0,Finanzas,es
Corporate Headquarters,CPCORP0,Finance,fr
Corporate Headquarters,CPCORP0,Sales,en
Corporate Headquarters,CPCORP0,Ventas,es
Corporate Headquarters,CPCORP0,Ventes,fr
```

## CSV Multilingual Field Support

### The Language Token

A specialized token that contains an instruction to apply on the column definition is added to a multilingual CSV column header. It is used to support the addition of new languages by adding multilingual field columns to an existing CSV file without having to modify the corresponding load file.

To use the CSV multilingual field support in TCC-TCB:

- The properties section of the CSV file **MUST** contain the "csv.multilingual=true" property. The property name is case sensitive and **MUST** be lowercase.
- The CSV header **MUST** be present. It means that the load file **MUST** have "Header present" checked, and that the first row of the CSV input file data is the header.
- All the CSV file header names **MUST** match the column path or the column alias (header) specified in the load file.

To add new multilingual field columns, at least one column of this multilingual field **MUST** already exist in the load file. The new column header added to the CSV file **MUST** be placed after all the existing column headers defined in the load file for that field. It is not mandatory to put the new header right after the last existing header, it could be placed at the end of the row.

To specify a locale, the token [locale=xx] **MUST** be used at the end of the header name (where xx is a specified locale for example "fr"). The "locale" token is case sensitive and **MUST** be lowercase.

### Example

Original CSV file:

```
Identifier,"UserLoginname","Title","ManagerLoginname"
abc123,abc123,architect,manager123
```

CSV file with additional multilingual fields:

```
#BEGINPROPERTIES
csv.multilingual=true
....
#ENDPROPERTIES
Identifier,"UserLoginname","Title","Title [locale=fr]",
"Title [locale=es]","ManagerLoginname"
abc123,abc123,architect,architecte,arquitecto,manager123
```

**Note:** The locale information is optional for the initial multilingual field columns defined in the load file.

- When the locale is NOT specified in the CSV file, the locale information is retrieved from the load file.
- When the locale is specified in the CSV file and the "csv.multilingual=true" property is set, it always overrides the locale information from the load file.

CSV file using specific locale:

```
#BEGINPROPERTIES
csv.multilingual=true
...
#ENDPROPERTIES
Identifier,"UserLoginname","Title [locale=fr]","ManagerLoginname"
abc123,abc123,architecte,manager123
```

## Internal Load File

When TCC loads a set of additional multilingual columns, it creates an internal load file that matches the structure of the CSV file. The attribute values for the new columns are copied from the first multilingual column defined in the load file. The locale token value sets the locale attribute value for each new column.

Original Load File

```
<load:column parameter="1" type="DATA" path="CorrespondenceEmail" ignoreEmpty="true" localeType="NONE"/>
<load:column parameter="1" type="DATA" path="Title" ignoreEmpty="true" localeType="DEFAULT"/>
<load:column parameter="1" path="Department,CostCenter" searchType="NONE" localeType="DEFAULT" type="DATA"/>
```

New CSV File

```
#BEGINPROPERTIES
csv.multilingual=true
#ENDPROPERTIES
CorrespondenceEmail,Title,"Title [locale=fr]",
"Department,CostCenter","Department,CostCenter [locale=fr]"
abc123@xyz.com,plumber,plombier,plumbing,plomberie
```

Internal Load File

```
<load:column parameter="1" type="DATA" path="CorrespondenceEmail" ignoreEmpty="true" localeType="NONE"/>
<load:column parameter="1" type="DATA" path="Title" ignoreEmpty="true" localeType="DEFAULT"/>
<load:column parameter="1" type="DATA" path=" Title" ignoreEmpty="true" localeType="SPECIFIC"
localeValue="fr"/>
<load:column parameter="1" path="Department,CostCenter" searchType="NONE" localeType="DEFAULT" type="DATA"/>
<load:column parameter="1" path="Department,CostCenter" searchType="NONE" localeType="SPECIFIC"
localeValue="fr" type="DATA"/>
```

Note that only the locale attributes are different for the new columns, all the others are the same.

## 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 within the RSOFFER module.
- Cannot request for offer approval using TCC. It has to be a manual process.

## Temporary File Compression

The compression feature for temporary files is now activated by default. The compressed temporary files now have a "zip" extension instead of a "tmp" extension.

This feature:

- Reduces the risk of reaching the 4 gigabyte maximum file size limitation due to the 32 bit JDK.
- Increases communication speed between each work-flow step.
- Saves disc space.

You can deactivate the compression mode by adding the following parameter in your TaleoConnectClient.ini file available under your TCC installation path.

```
-Dcom.taleo.integration.compress.tempfile=false (or FALSE)
```



# 6 Net Change

## Description

### Operation

The Net Change operation extends the functionality of TCC by allowing the customer to reduce the amount of transactions that are required to synchronize his system with the Taleo Enterprise application.

### Purpose

The Net Change operation increases the update process efficiency when it's impossible to send only the required changes to a Taleo application, and the only alternative is to send all the data for a certain type of entity.

Net Change checks the data being sent, and removes all unchanged data from the transmission. This improves the integration process by limiting the amount of data applied to the database and making it easier to respect the volume limits.

### Process

Net Change uses a flat file proprietary database system to store the data that is equivalent to the target application (e.g. Taleo Recruiting). The Net Change tables are created on an Entity by Entity basis on first time use whether it be Net Change Setup or ongoing run. This information is used to synchronize customer system and target application.

The customer's data must be provided in a standard TCC CSV file. The structure of this file must be created with the TCC Import Wizard. The customer selects the entity on which to apply the Net Change process and builds the CSV file structure by selecting the entity fields available in the entity data model. Standard Net Change templates are available.

The Net Change work-flow is defined in a standard TCC configuration file.

To start a Net Change process, the TCC execution mode (GUI or Command Line) must be invoked. The customer data CSV file is retrieved and processed by Net Change. At this point, the customer data is compared to the Net Change database and transactions are generated for all the differences that are detected. The output of this computation is a T-XML file containing all the transactions required to update the target application. The T-XML file is then wrapped in a SOAP envelope and is sent to the target application.

TCC polls the target application for an import request completion, and retrieves the result document when it's ready. The result processor output is a SOAP wrapped T-XML file that contains Net Change transaction status and attributes for each entity processed. The T-XML output file is converted to a standard TCC CSV response document.

The Net Change Result Processor validates the answer for each transaction, and those successfully processed by the target application are committed to the Net Change database. Transactions reported in error are not committed and will be reprocessed on the next Net Change run. The entities are now synchronized in application and customer databases.

### Primary-Secondary Concept

When synchronizing customer and target application data with Net Change, the Primary-Secondary concept must always be respected. The customer data is the Primary and the modified data replicated to the target is the Secondary.

Modifications must always be done in the Primary (CSV file). The modified data replicated to the target (Secondary) with Net Change should not be modified in any way.

Certain business cases require that only some of the data be managed by Net Change in a Primary-Secondary relation, while the target application manages other data known only to itself. Candidate entities give us a good example. There are two types of candidates, external and internal. Internal candidates are managed by the customer system (Primary) and the external candidates by the Taleo Enterprise system (Secondary). Only the internal candidates need to be synchronized with the target application.

## Recruiting Entities

### Candidate

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Inactivate. Invokes entity update operation to set Internal Application to 'N' as specified in the CSV file.
- Recover: Reactivate. Invokes entity merge operation to set Internal Application to 'Y' as specified in the CSV file.

### Department

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Inactivate. Invokes the entity update operation to set the Department StateNumber to '1' (Inactive).
- Recover: Reactivate. Invokes entity merge operation to set the Department StateNumber to '0' (Active).

Department is a hierarchical entity.

Hierarchical entities require a complete Net Change Setup prior to regular usage of Net Change.

Hierarchical entities require that you select "Skip column if value is empty" for the parent column. Otherwise it will not be possible to import the root element because its parent column will be empty.

### Job Template

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Inactivate. Invokes the entity update operation to set the Job Template State Number to '3' (Inactive).
- Recover: Reactivate. Invokes the entity merge operation to set the Job Template State Number to '1' (Active).

### LUDS (UDSElement)

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Inactivate. Invokes the entity update operation to set the UDS Element Active attribute to false (Inactive).

- Recover: Reactivate. Invokes the entity merge operation to set the UDS Element Active attribute to true (Active).

The LUDS (UDSElement) Net Change import file (load file), must have two columns defined on the same path to work correctly.

The first column defines the User Defined Selection (UDS) that contains the UDS Element. This is required because the key is UDS+UDS Element.

- The column contains "UserDefinedSelection,Code".  
Check "Use the value to search or lookup".  
Select UDSElement as Target.

The second column defines to which UDS the UDS Element belongs to.

- The column contains "UserDefinedSelection,Code".  
Check "Use the value to search or lookup".  
Select UserDefinedSelection as Target.  
Check "Start a new element in the collection".  
Select UserDefinedSelection as Target.

## SmartOrg Entities

### Employee

- Create: First invokes the recover operation then the merge operation. The Employee is created if it does not exist, or updated if it exists.
- Update: Invokes entity merge operation.
- Delete: Invokes the delete operation that performs a soft delete of the Employee.
- Recover: First invokes the recover operation. Once the Employee is recovered, it invokes the merge operation to update the Employee.

Employee is a hierarchical entity.

Hierarchical entities require a complete Net Change Setup prior to regular usage of Net Change.

Hierarchical entities require that you select "Skip column if value is empty" for the parent column. Otherwise it will not be possible to import the root element because its parent column will be empty.

### Group

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Invokes entity delete operation.
- Recover: No recover on hard delete.

## JobField

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Invokes entity delete operation.
- Recover: No recover on hard delete.
- Special Processing: Switches the zone into maintenance mode, if not already so, to perform create, update, and delete operations.

JobField is a hierarchical entity.

Hierarchical entities require a complete Net Change Setup prior to regular usage of Net Change.

Hierarchical entities require that you select "Skip column if value is empty" for the parent column. Otherwise it will not be possible to import the root element because its parent column will be empty.

## Location

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Invokes entity delete operation.
- Recover: No recover on hard delete.
- Special Processing: Switches the zone into maintenance mode, if not already so, to perform create, update, and delete operations.

Location is a hierarchical entity.

Hierarchical entities require a complete Net Change Setup prior to regular usage of Net Change.

Hierarchical entities require that you select "Skip column if value is empty" for the parent column. Otherwise it will not be possible to import the root element because its parent column will be empty.

## Organization

- Create: Invokes entity merge operation.
- Update: Invokes entity update/move operation. Update can generate up to two transactions, one to update and one to move. Update invokes entity update operation for changes in fields other than Parent field. For changes detected in Parent field, Update invokes move operation to link Organization to different Parent.
- Delete: Invokes entity delete operation.
- Recover: No recover on hard delete.
- Special Processing: Switches the zone into maintenance mode, if not already so, to perform create, update, and delete operations.

Organization is a hierarchical entity.

Hierarchical entities require a complete Net Change Setup prior to regular usage of Net Change.

Hierarchical entities require that you select "Skip column if value is empty" for the parent column. Otherwise it will not be possible to import the root element because its parent column will be empty.

## Position

- Create: Invokes entity merge operation.
- Update: Invokes entity update operation.
- Delete: Inactivate. Invokes entity update operation to set the Position Active attribute to false (Inactive).
- Recover: Reactivate. Invokes entity merge operation to set the Position Active attribute to true (Active).

## User

- Create: First invokes the recover operation then the merge operation. The User is created if it does not exist, or updated if it exists.
- Update: Invokes entity update operation.
- Delete: Invokes entity delete operation that performs a soft delete of the User.
- Recover: First invokes the recover operation. Once the User is recovered, it invokes the merge operation to update the User.

## Creating a Net Change Integration

1. Create a TCC import script.
2. Create the Base File that contains the complete data set for the specified columns in the import script.
3. Create a TCC Net Change Setup configuration file using the TCC import script.
4. Run the TCC script to perform the Net Change Setup.  
This creates and populates the Net Change table locally on the computer where TCC is installed.
5. Create a TCC Net Change Computation configuration file using the same TCC import script.
6. Run the TCC script to perform the Net Change Computation.  
This will update the Net Change table and the Taleo database.

## Import Files

### CSV and Load Files

Net Change requires an import definition load file and a CSV data file to function. These two files are used by Net Change for both the Setup and Computation runs.

#### CSV File

Net Change expects a TCC CSV file as input. The file must contain all customer data for an entity. It is:

1. Used as a reference to create the Net Change database during Setup.
2. Compared to the Net Change database to detect any changes during Computation.

## Load File Creation

Use the Import Wizard to create the Net change load file that will define the CSV file structure.

**New Import Wizard**

This wizard creates a new import specification by selecting a service operation from one of the available product models.

**Select a service**

Product : Smartorg

Model : Smartorg

Filter :

IntegrationCustomField

**JobField**

JobRole

Location

MatrixManager

**Template**

Create a new import specification or use a template

Create new import specification

Operation : net-change

Use the following import template

Template :

Finish Cancel

Select the **net-change** operation. This operation is only available for Net Change supported entities.

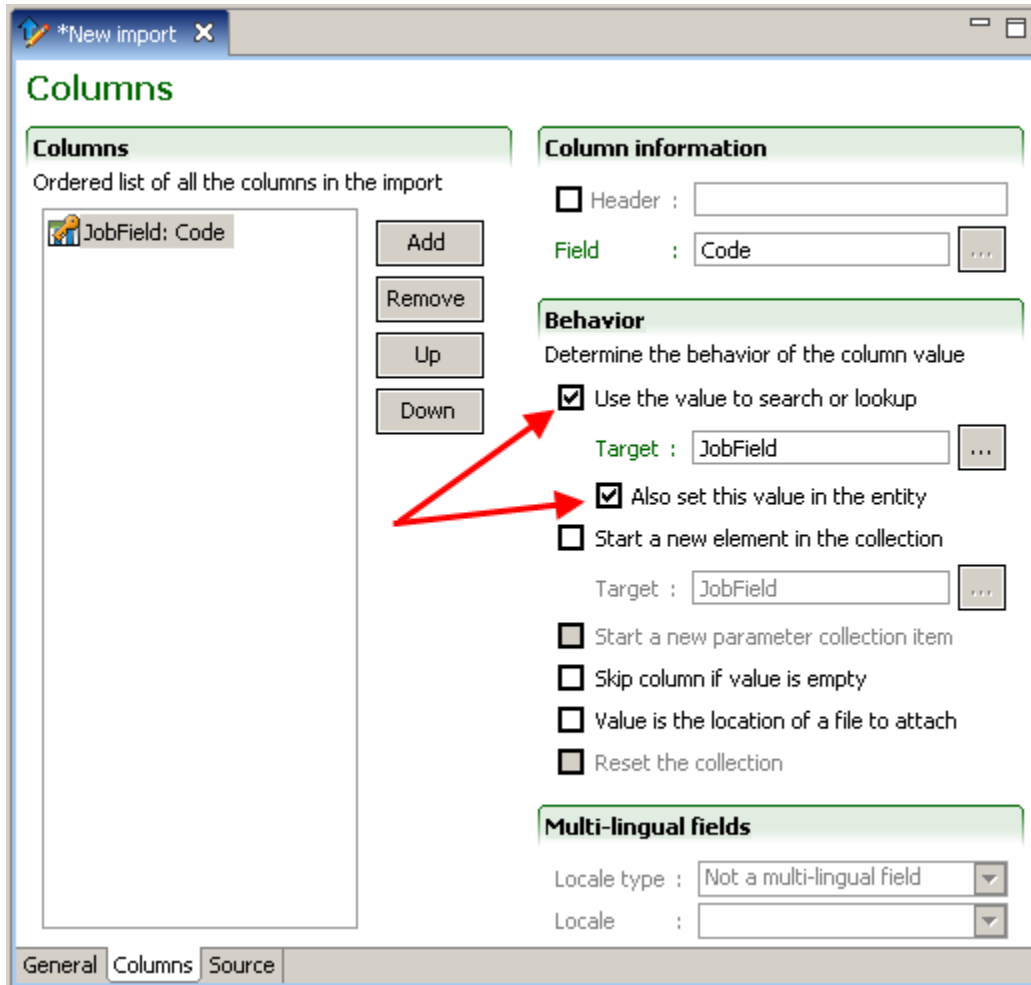
## Column Import Options

### Entity Key Column

An entity must have a search key. A key can be composed of one or more fields. This key is used to retrieve corresponding information in the Net Change database.

In the Behavior section, check “*Use the value to search or lookup*” and “*Also set this value in the entity*” to create a new entity or to update an existing entity. If “*Also set this value in the entity*” is not checked for a create operation, certain

entity types will create records without a known key value, and it will then be impossible to search for these records to update them afterwards. For other entity types, the records are not created and an error message is returned.



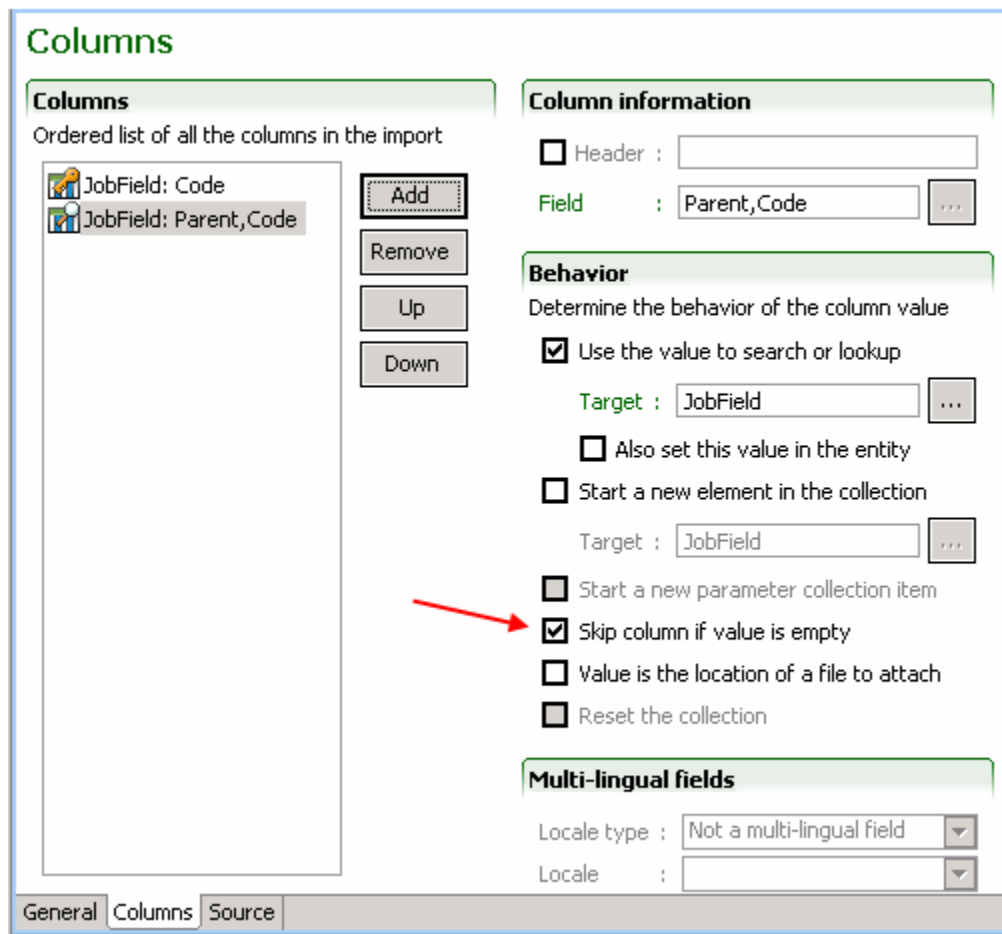
## Collections

Do not check “Reset the collection” and “Remove element from the collection”. The Net Change process is designed to manage collection synchronization automatically.

## Skip column if value is empty

To prevent the processing of fields where no value is provided, select *Skip column if value is empty* in the Behavior section.

Hierarchical entities such as Organization, Location, JobField, and Department, where root elements do not have a value in the parent column will be processed anyway.



## Collection handling

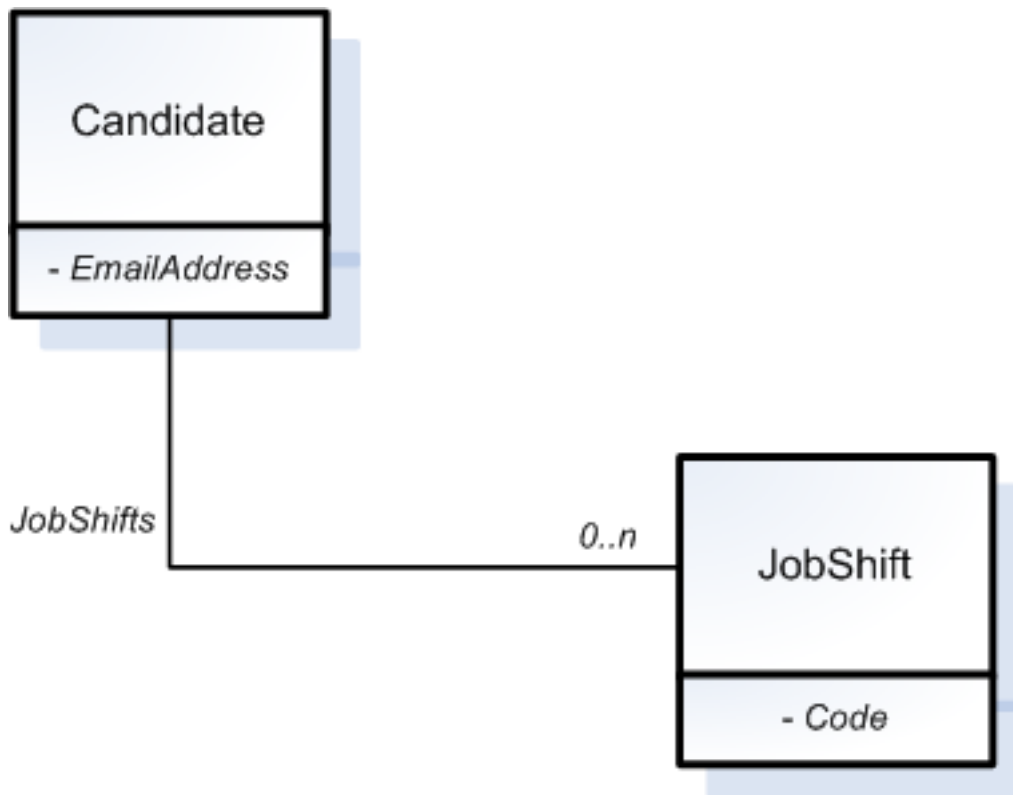
When using the *Skip column if value is empty* option on a collection, it is important to understand and apply the following rules:

1. In the CSV file, the collection elements are not positional. Therefore it makes no sense to set the *Skip column if value is empty* option only on some columns of the collection.
2. The *Skip column if value is empty* option applies to the entire collection. It does NOT apply to an individual element of the collection.
3. A collection is empty when ALL its elements are empty.

## Simple collection

A `candidate` who has a collection of `jobshifts`.





Let say that in the CSV file we support up to three `JobShifts`:

| Candidate, EmailAddress | JobShift 1. Code | JobShift 2. Code | JobShift 3. Code |
|-------------------------|------------------|------------------|------------------|
| cand_001@abc.com        | DayJob           | OnCall           |                  |

**Rule 1:**

In the CSV file, the collection elements are not positional. Therefore it makes no sense to set *Skip column if value is empty only* on some columns of the collection.

The following CSV file has exactly the same meaning as the one above:

| Candidate, EmailAddress | JobShift 1. Code | JobShift 2. Code | JobShift 3. Code |
|-------------------------|------------------|------------------|------------------|
| cand_001@abc.com        | OnCall           | DayJob           |                  |

The point here is the DayJob value is not associated to the JobShift1.Code column. It is only an element of the JobShifts collection. The Net Change engine will load all the elements of the collection and then it will compare them with the one in its database.

**Rule 2**

The *Skip column if value is empty* option applies to the entire collection. It does NOT apply to an individual element of the collection.

First run:

| Candidate, EmailAddress | JobShift 1. Code | JobShift 2. Code | JobShift 3. Code |
|-------------------------|------------------|------------------|------------------|
| cand_001@abc.com        | DayJob           | OnCall           |                  |

The engine will add the DayJob and OnCall job shift values in the `JobShifts` collection.

Second run:

| Candidate, EmailAddress | JobShift 1. Code | JobShift 2. Code | JobShift 3. Code |
|-------------------------|------------------|------------------|------------------|
| cand_001@abc.com        | DayJob           |                  |                  |

Even if the OnCall job shift is no longer in the collection (empty) the entire collection in itself is not empty. The Net Change engine will generate a command to remove the OnCall job shift from the `JobShifts` collection.

### Rule 3

A collection is empty when ALL its elements are empty.

The *Skip column if value is empty* option will be applicable only in the following scenario:

First run:

| Candidate, EmailAddress | JobShift 1. Code | JobShift 2. Code | JobShift 3. Code |
|-------------------------|------------------|------------------|------------------|
| cand_001@abc.com        | DayJob           | OnCall           |                  |

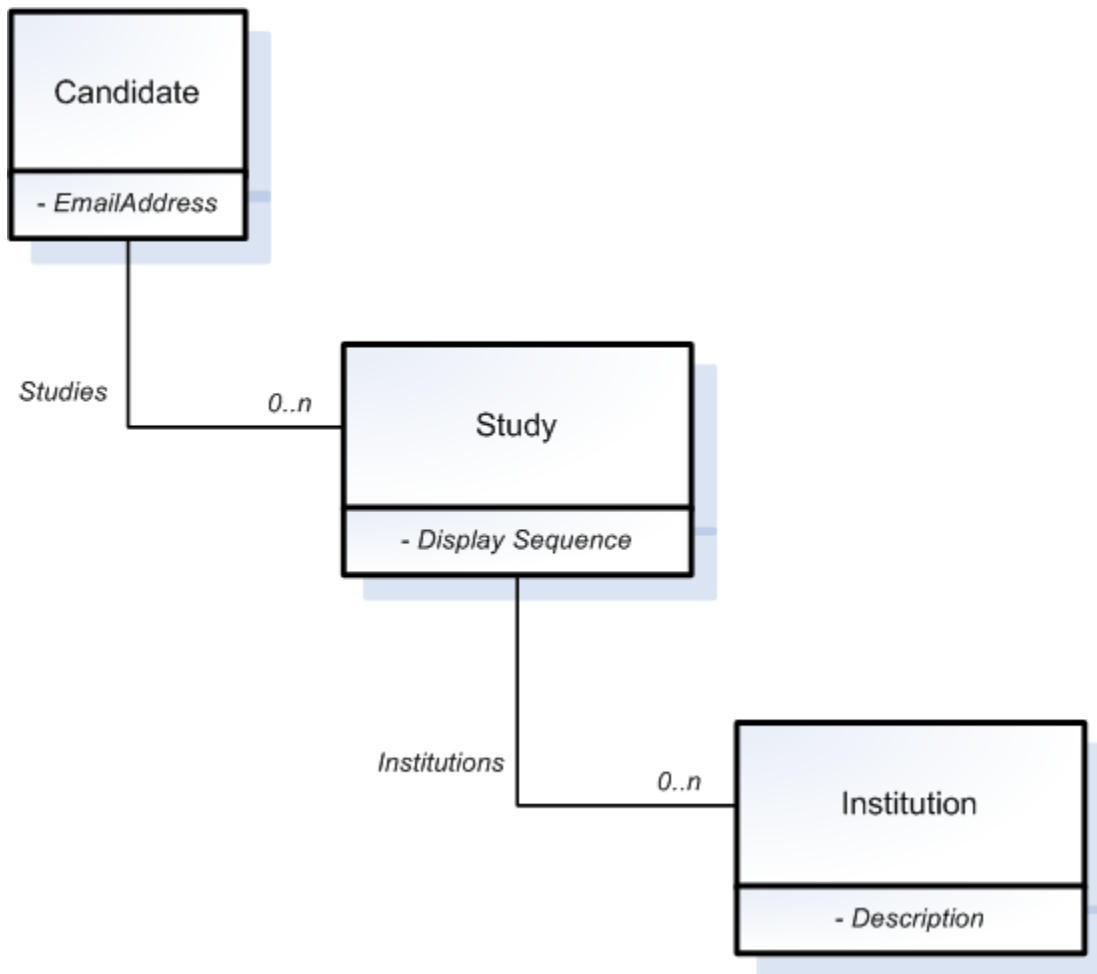
DayJob and OnCall are added in the `JobShifts` collection.

Second run:

| Candidate, EmailAddress | JobShift 1. Code | JobShift 2. Code | JobShift 3. Code |
|-------------------------|------------------|------------------|------------------|
| cand_001@abc.com        |                  |                  |                  |

When the *Skip column if value is empty* option is set of the JobShift1.Code, JobShift2.Code and JobShift3.Code columns, then the empty values are ignored (skipped).

### Complex collection



Let say that in the CSV file we support two `studies`:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        | 1                        | College A                        | 2                        | University Z                     |

**Rule 1**

In the CSV file, the collection elements are not positional. Therefore it makes no sense to set *Skip column if value is empty* only on some columns of the collection.

The following CSV file as exactly the same meaning as the one above:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        | 2                        | University Z                     | 1                        | College A                        |

### Rule 2

The *Skip column if value is empty* option applies to the entire collection. It does NOT apply to an individual element of the collection.

First run:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        | 1                        | College A                        | 2                        | University Z                     |

The engine will add the 1, College A and 2, University Z study to the `studies` collection.

Second run:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        | 1                        | College A                        |                          |                                  |

Even if the 2 - University Z values are no longer in the collection (empty) the entire collection in itself is not empty. The Net Change engine will generate a command to remove the 2 – University Z study element from the `studies` collection.

### Rule 3

A collection is empty when ALL its elements are empty.

With the following collection, the *Skip column if value is empty* option will be applicable only in the following scenario:

First run:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        | 1                        | College A                        | 2                        | University Z                     |

1, College A and 2, University Z are added to the candidate `studies` collection.

Second run:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        |                          |                                  |                          |                                  |

If the *Skip column if value is empty* option is set on ALL the columns of the `studies` collection (Study1.DisplaySequence, Study1.Institution.Description, Study2.DisplaySequence and Study2.Institution.Description), then the empty values are ignored (skipped).

The following collection is not considered to be empty:

| Candidate. EmailAddress | Study1. Display Sequence | Study1. Institution. Description | Study2. Display Sequence | Study2. Institution. Description |
|-------------------------|--------------------------|----------------------------------|--------------------------|----------------------------------|
| cand_001@abc.com        | 1                        |                                  |                          |                                  |

**Note:** If you set the "Skip column if value is empty" option to only one element of the collection, the Net Change engine will automatically set it on the other elements of the collection. So take the good habit of setting that option on all the elements of the collection the same way.

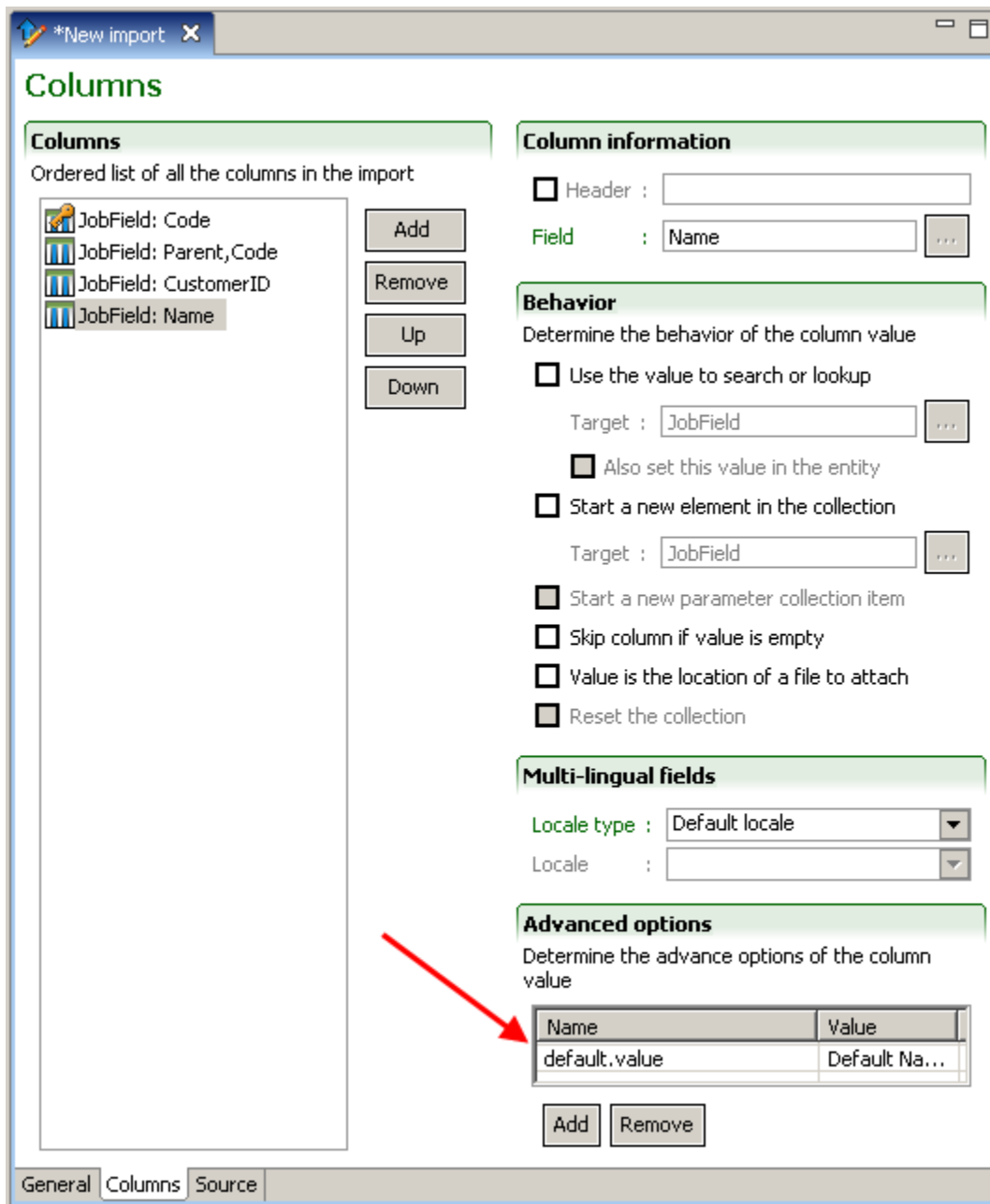
## Ignore Column

When the customer system has a column that is known and used only by itself. An ignore column that is never processed can be specified. To define, click Add button, select "Add an ignore column", and click OK.

## Default Column Value

A default value can be specified for columns that have no value set in the customer CSV file. For example, to use English as default value for field `User.CorrespondenceLanguage`, set the default value to "en" in the column advanced options:

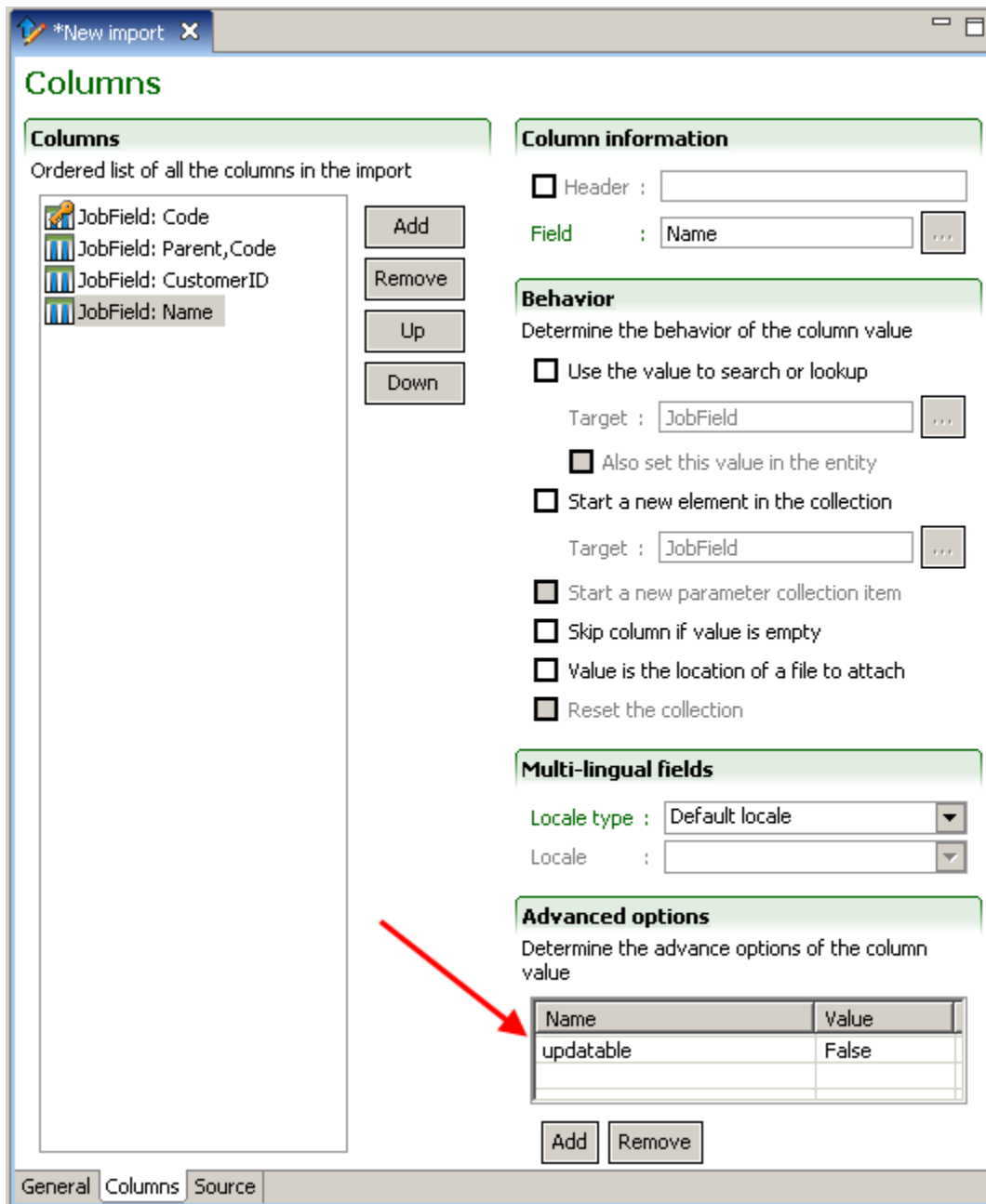
Note that "Default Column Value" processing is not done when "Skip column if value empty" is checked.



## Non-Updatable Column

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. No other TCC functionality uses it. Set updatable value to false in the column advanced options:



When all columns are defined, save the input file and generate the CSV file.

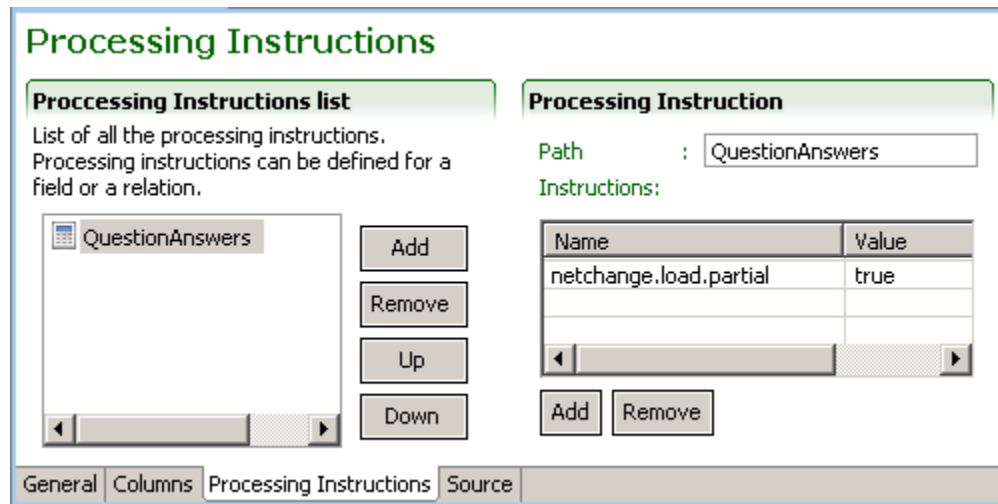
## Net Change Partial Load

The Net Change partial load processing instruction limits the extraction of collection type relation items to those specified in the customer data file during Net Change Setup. This subset of relation items is used to populate the Net

Change database. This allows customers to work with partial collection data sets without creating deletes in their zone with ongoing Net Change 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.

## Multiple Parameter Entities

The User, Organization, Location, and JobField entity import services have more than one parameter.

During Net Change CSV load file creation, it is mandatory that every available parameter of the service be included in the file. Not doing so will cause Net Change to fail.

The User import service exposes three parameters:

1. User
2. Password
3. ForceChangePassword

You must include the Password and ForceChangePassword simple parameters, and at least one field from the User entity parameter in the CSV file.

The Organization, Location, and JobField import services expose two parameters:

1. Organization, Location, or JobField
2. Sequence

You must include the Sequence simple parameter, and at least one field from the Organization, Location, or JobField entity parameter in the CSV file.

In the Behavior section check “Skip column if empty” for the columns that might not contain a value.



## Hierarchical Trees

Net Change must apply a set of special processing rules to entities that are part of a hierarchical tree. For example, before deleting a parent node, all of its children must have been deleted. Net Change recreates the entire hierarchical tree to correctly sequence the order of operations required to synchronize the target application.

Hierarchical Entities:

- Department
- Employee
- JobField
- Location
- Organization

There are two modes for hierarchical entities, full and partial. The full mode allows only a single root element, while the partial mode allows multiple root elements. By default the mode is full for hierarchical entities. The Employee entity can be full or partial.

The customer input CSV file must contain the entire data set of the hierarchical tree. The first line of the CSV file must be the root element, and will have no parent. In the Parent column, make sure “Skip column if value is empty” is checked. For partial mode, the root elements must precede their children in the CSV file. See partial example below.

Full

```
"Code (Child)", "Parent,Code", "Description", ...
"-1", "Organization", ...
"DIV-A", "-1", "Division A", ...
"DIV-B", "-1", "Division B", ...
"T1", "DIV-A", "Team 1", ...
"T2", "DIV-A", "Team 2", ...
"T3", "DIV-B", "Team 3", ...
```

Partial

```
"Code (Child)", "Parent,Code", "Description", ...
"-1", "Boss", ...
"MNG-A", "-1", "Manager A", ...
"E1", "MNG-A", "Employee 1", ...
"E2", "MNG-A", "Employee 2", ...
"MNG-B", "Manager B", ...
"E3", "MNG-B", "Employee 3", ...
```

## Errors

Hierarchical entities must be handled properly to prevent errors. The three main causes that generate errors using Net Change with hierarchical entities (HE) are:

1. Using a parent relation that doesn't exist.

Each time you create an HE that refers to a parent that doesn't exist in Monarch, you will get this kind of error.

Monarch Functional Error: The parent node associated to this key has not been found: ...

2. Working with a partial “tree”.

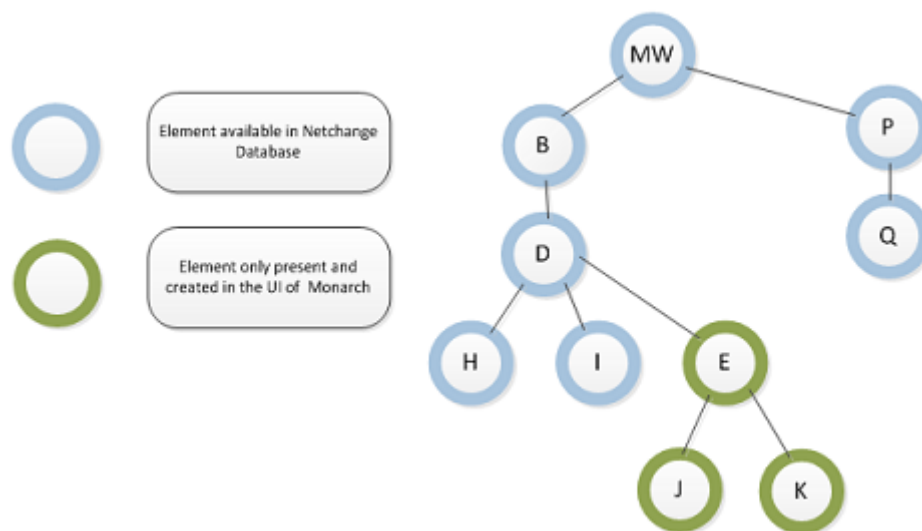
Each time you create an HE that refers to a parent that doesn't exist in partial "tree" but exists in Monarch, you will get this kind of error.

Monarch Functional Error: The parent node associated to this key has not been found: ...

3. Transactions done directly in the Monarch application.
  - a. We can generate an invalid structure in "Net Change database" and destroy the "Net Change process" when we are doing modification in the UI of Monarch (Element in Green) while doing Net Change with TCC.

If we do a "Net Change Setup" action using an existing element (in Monarch, Element J) that is not directly linked to an element of the partial "tree" (used in "Net Change"), the "tree" is not valid anymore.

- b. If we do a "Net Change Setup" action using an existing element (in Monarch, Element E) that is directly link to an element of the partial "tree" (used in the "Net Change"), the "tree" is valid and it's possible to do Net Change if we do a "Setup" first and then a "Computation".



### Examples:

- Case-1: Do Net Change Computation only (Move "Element J" under "Element B").  
Monarch Functional Error: ...You cannot update the following field: BaseComposite.Parent.  
Note that the field value and the supplied value don't match (as described in 3-a).
- Case-2: Do Net Change Setup & Computation (Move "Element J" under "Element B").
  - Setup runs with success.
  - Computation TCC WorkFlow Error: The parent node associated to this key has not been found .... (as described in 3-a).
- Case-3: Do Net Change Computation only (Move "Element E" under "Element B").  
Monarch Functional Error: ...You cannot update the following field: BaseComposite.Parent.

Note that the field value and the supplied value don't match (as described on 3-b).

- Case-4: Do Net Change Setup & Computation (Move “Element E” under “Element B”).
  - Setup runs with success.
  - Computation runs with success (as described on 3-b).

The Partial hierarchical entity “Employee” is not impacted by the behavior described above. When doing Net Change with an Employee, you can get an empty parent or a parent that is not present in your Net Change Database.

We recommend that you always work with a full structure and respect the “Primary/Secondary” concept when doing your Net Change otherwise you can end up with a Employee structure that does not represent your company.

## CSV File Properties

Special header properties can be added to customize Net Change processing behavior. They are placed at the beginning of the CSV file.

CSV Properties section:

```
#BEGINPROPERTIES
PropertyName=PropertyValue
#ENDPROPERTIES
```

### CSVLineCount

This property indicates to Net Change the number of data rows in the request CSV file (excluding the column header row). A mismatch is considered an error.

Example:

```
#BEGINPROPERTIES
CSVLineCount=2
#ENDPROPERTIES
ColumnHeader1,ColumnHeader2,ColumnHeader3
Line1_CH1_Data,Line1_CH2_Data,Line1_CH3_Data
Line2_CH1_Data,Line2_CH2_Data,Line2_CH3_Data
```

## Import File Modification

Net Change import file modification may or may not have any impact on Net Change. Here are the possible cases:

### Entity Search Key Modification

If the defined entity search key must be changed after the first Net Change run, the Net Change database for this entity must be recreated. This can be achieved using the Net Change Setup feature.

### Adding a new column

Net Change sees the new column as a new entity field.

When “*Skip column if value is empty*” is NOT checked, an update is generated for every record.

When “*Skip column if value is empty*” is checked, an update is only generated for records that have a value for the new column.

Note that only the fields that are detected as changed are sent to the target application.

## Removal of a column

If a column is removed, the data associated to this column is no longer part of the Net Change computation process and no actions/updates are sent to the zone.

## Changing the column from Updatable to Non-Updatable

The next Net Change computation does not take into account the value of this column for an update operation.

# Configuration Files

## Setup and Computation Configuration Files

Net Change requires a Setup and a Computation configuration file for each individual integration point. The Setup configuration file is used to create and populate the Net Change database. The Computation configuration file is used for Net Change ongoing runs. The differences between the files are found in the pre and post processing steps.

## Configuration File Creation

Use the Configuration Wizard to create the Net change configuration file.

Select *Based on another type of request message* and the product for that integration point.

All communications with the target application use T-SOAP.

Select T-SOAP in the message type selection window.

Select Endpoint to authenticate yourself to the target application.

## General Tab

Net Change requires that both the request and response files be of CSV type.

## Pre-processing Steps

### Setup

Add a Net Change Setup custom step.

Select the load file to use as import file for this integration. This is the same file as the one used for Computation.

Add a Wrap SOAP step

Select Import in the Action panel.

## Computation

Add a Net Change Computation custom step.

Select the load file to use as import file for this integration. This is the same file as the one used for Setup.

### Thresholds

You can define thresholds, to limit the number of actions that Net Change applies to the target application. For example, in normal operation mode, the generation of delete actions deleting almost all of the target application database should be considered as an erroneous situation and should be prevented.

The threshold can be defined as a percentage or a number of records. For example a value of 25% stops the Net Change computation and process if more that 250 records out of a 1000 are generating Net Change actions.

An absolute value can be specified that will stop the Net Change computation and process as soon as the number of actions is reached. The number of record in the file has no influence on this threshold. A file of 50 records or 1000 records will stop after 25 processed records.

To disable the threshold, set the value to -1.

A threshold must be defined for each Net Change action: Create, Update and Delete. Recommended values for thresholds are as follow:

- Threshold for element creation: 10%
- Threshold for element update: 10%
- Threshold for element deletion: 10%

Add a Wrap SOAP step

Select Import in the Action panel.

## Post-processing Steps

### Setup

Add a Net Change Result Processor custom step.

### Report Filtering

Filters are available to control the output of the Net Change result file:

- **All:** Detailed status of all actions are reported.
- **Error only:** Detailed status of failed actions are reported.
- **Error and warning only:** Detailed status of actions that have generated an error or a warning message are reported.

When performing a Net Change Setup, this step generates a simple XML report that indicates how many entities have been processed during the operation. Since a Net Change Setup does not import any data into the product, there are no entity results in this report.

Add a Net Change Result Processor custom step.

This step is optional for Net Change Setup. It transforms the file generated by the *Net Change Result Processor* into a standard TCC CSV file.

Select the load file to use as import file for this integration. This is the same file as the one used for Computation.

Select the Output properties check box to include the CSV file properties in the result file.

| Property       | Description  | Example              |
|----------------|--|----------------------|
| CSVLineCount   | Number of data rows present in the CSV result file (excluding the column header row). In the case of an Net Change Setup process, this value will always be 0.   | CSVLineCount=0       |
| total.[entity] | Total number of entities processed. If no entities are processed by the target application, this property is not included.   | total.jobfield=96    |
| duration       | Time taken by target application to process entire import file generated by Net Change engine. In the case of an Net Change Setup, no entities are imported in the customer zone, the duration should be close to 0. | duration=0:00:00.106 |

### CSV Result File Format

The file generated by Net Change Result Transformation has the following format (Note that if in the configuration general tab you have specified to not output the column header, then no header will be included in the result CSV file):

```
#BEGINPROPERTIES
PropertyName=PropertyValue
#ENDPROPERTIES
"Index", "Identifier", "Status", "TransactionType", "Result", "Message"
```

### CSV Result File Description

Net Change Setup processing does not generate any rows in the CSV file since no imports are done on the customer zone.

## Computation

Add a Net Change Result Processor custom step.

### Report Filtering

Filters are available to control the output of the Net Change result file:

- **All:** Detailed status of all actions are reported.
- **Error only:** Detailed status of failed actions are reported.
- **Error and warning only:** Detailed status of actions that have generated an error or a warning message are reported.

This step processes the target application result file to update the local Net Change reference database in order to reflect the state of the target application.

For each record, if the transaction was successfully processed, then the local Net Change reference database is updated with the new value. If the record was not successfully processed, the Net Change reference database is not updated.

Add a Net Change Result Processor custom step.

This step transforms the file generated by the *Net Change Result Processor* into a standard TCC CSV file.

Select the load file to use as import file for this integration. This is the same file as the one used for Computation.

Select the Output properties check box to include the CSV file properties in the result file.

| Property              | Description  | Example                  |
|-----------------------|--|--------------------------|
| CSVLineCount          | Number of data rows present in the CSV result file (excluding the column header row).  | CSVLineCount=96          |
| total.[entity]        | Total number of entities processed.<br>If no entities are processed by the target application, this property is not included.                                  | total.jobfield=96        |
| total.[entity].create | Total number of entities created.<br>If no entities are created, this property is not included.  | total.jobfield.create=48 |
| total.[entity].update | Total number of entities that are updated.<br>If no entities are updated, this property is not included.   | total.jobfield.update=10 |
| total.[entity].delete | Total number of entities that are deleted.<br>If no entities are deleted, this property is not included.   | total.jobfield.delete=10 |
| total.status.success  | Number of entities that are processed with success by the target application.<br>If no success has been detected, this property is included with a value of 0. | total.status.success=30  |
| total.status.error    | Number of entities reported in error at the target application.<br>If no error is reported, this property is included with a value of 0.                       | total.status.error=48    |
| total.warning         | Number of warning messages generated by the target application.<br>If no warning is generated, this property is included with a value of 0.                    | total.warning=18         |
| duration              | Time taken by target application to process entire import file generated by Net Change engine.   | duration=0:00:03.2890    |

### CSV Result File Format

The file generated by Net Change Result Transformation has the following format (Note that if in the configuration general tab you have specified to not output the column header, then no header will be included in the result CSV file):

```
#BEGINPROPERTIES
PropertyName=PropertyValue
#ENDPROPERTIES
"Index", "Identifier", "Status", "TransactionType", "Result", "Message"
```

| Column Name     | Description  |
|-----------------|--|
| Index           | Row sequential number  |
| Identifier      | <p>Unique key name used Net Change to identify the entity. This key is composed of the following sub sections:<br/>Entity Name Net Change Action Entity Search Key Id</p> <p>Example:<br/>JobTemplate create NC2_00791130771</p>   |
| Status          | <p>Net Change Result Status for the entity. Indicates whether Net Change operation completed with success or error.<br/>The possible values are:</p> <ul style="list-style-type: none"> <li>• success</li> <li>• error</li> </ul>  |
| TransactionType | <p>Target application transaction type to perform Net Change action.<br/>For example, if an update has been detected for the job template entity, the transaction type will be "jobtemplate.update".</p> <p>Note that the action done on the target application might be different than the Net Change action. For example, in the case of inactivation when a Net Change delete action is detected, an update operation is required. In this case the Net Change Action will be delete and the transaction type will be update.</p> |
| Result          | Not used for now.  |
| Message         | Detailed message about the error or the warning returned by the target application.  |

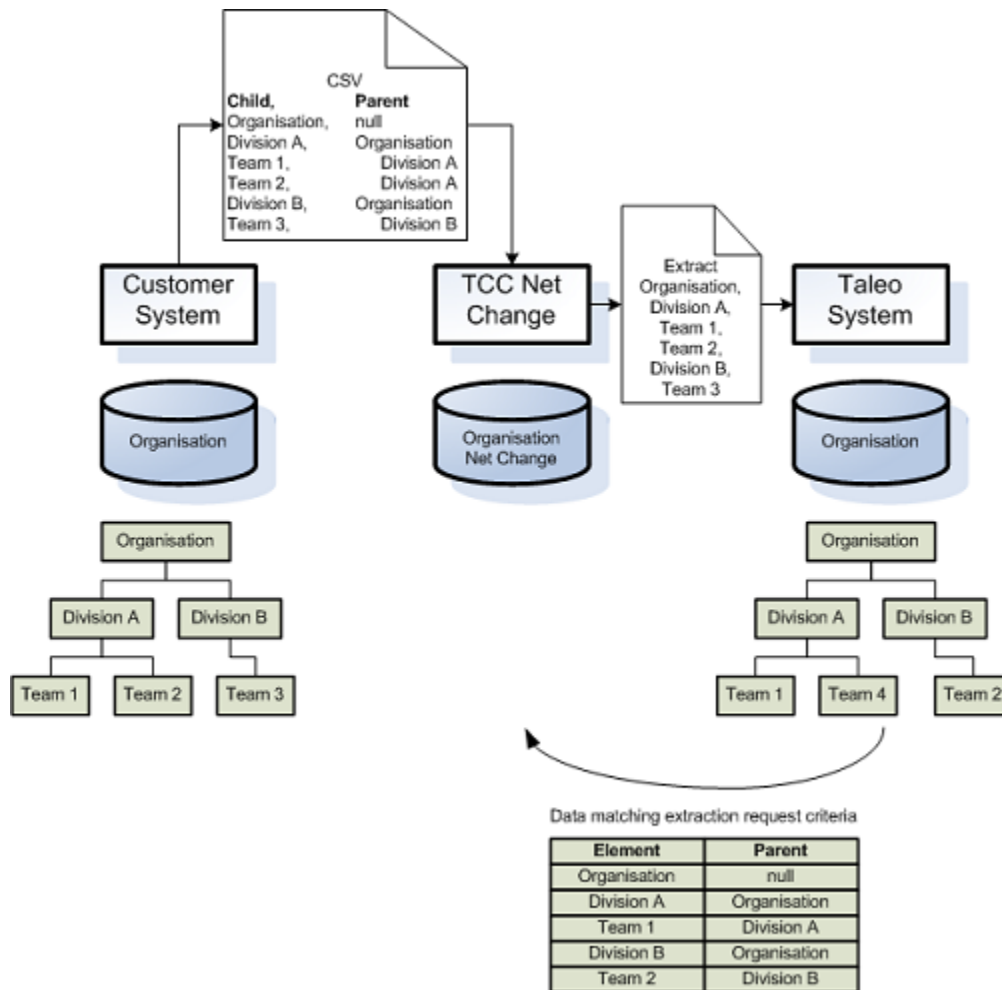
## Using Net Change

### Hierarchical Entities Setup

Special care must be taken with hierarchical entities during Net Change Setup. You must make sure to respect the hierarchical structure in order to maintain consistency.

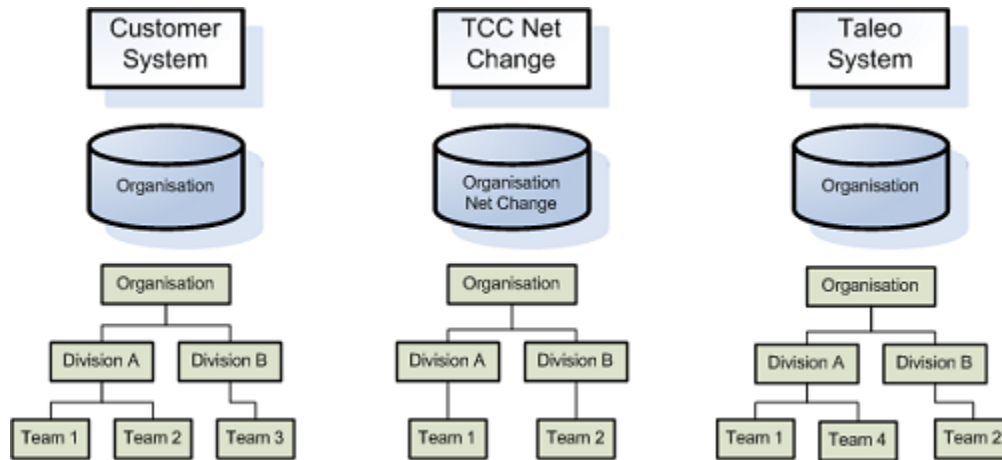


## Extract Net Change Setup Data



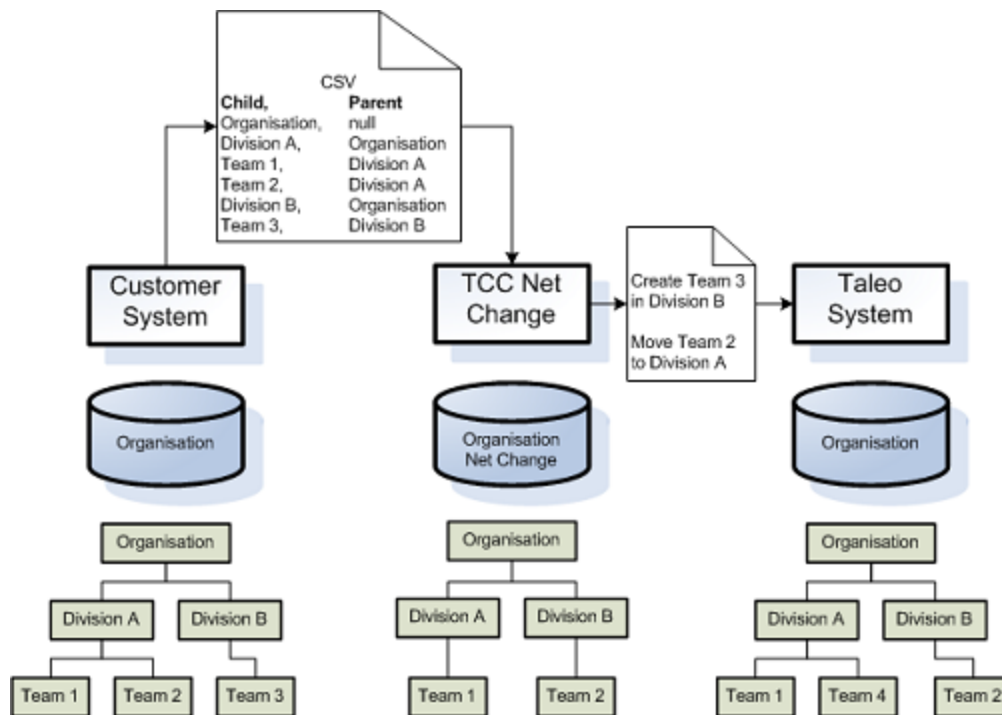
Net Change Setup extracts from the target application data matching the data keys present in the customer CSV file and fills the Net Change database. Other data, not specified in the CSV file that exist in the target application is not downloaded to the Net Change database.

## After Net Change Setup

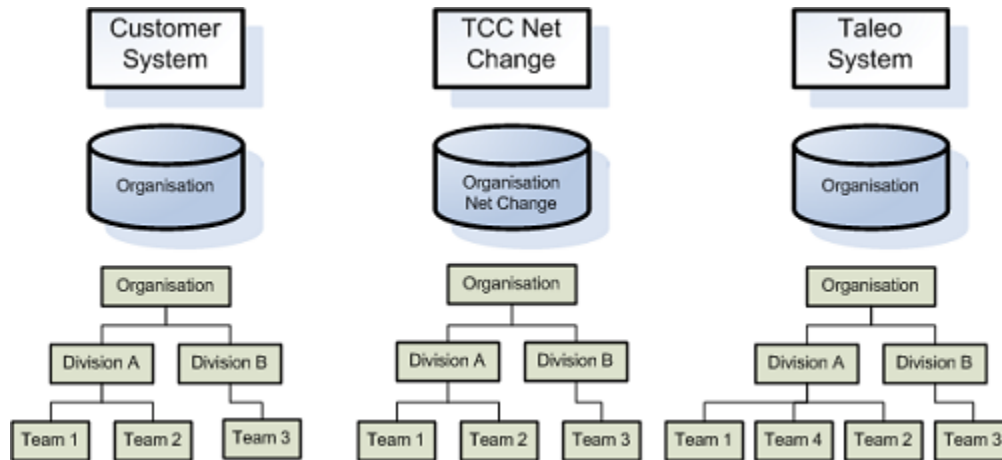


After a Net Change Setup, a standard Net Change process can be started to synchronize customer data with the target application. Data that is the same does not generate Net Change transactions. Different data generates transactions, customer data that does not exist in the target application generates a create transaction.

## First Net Change

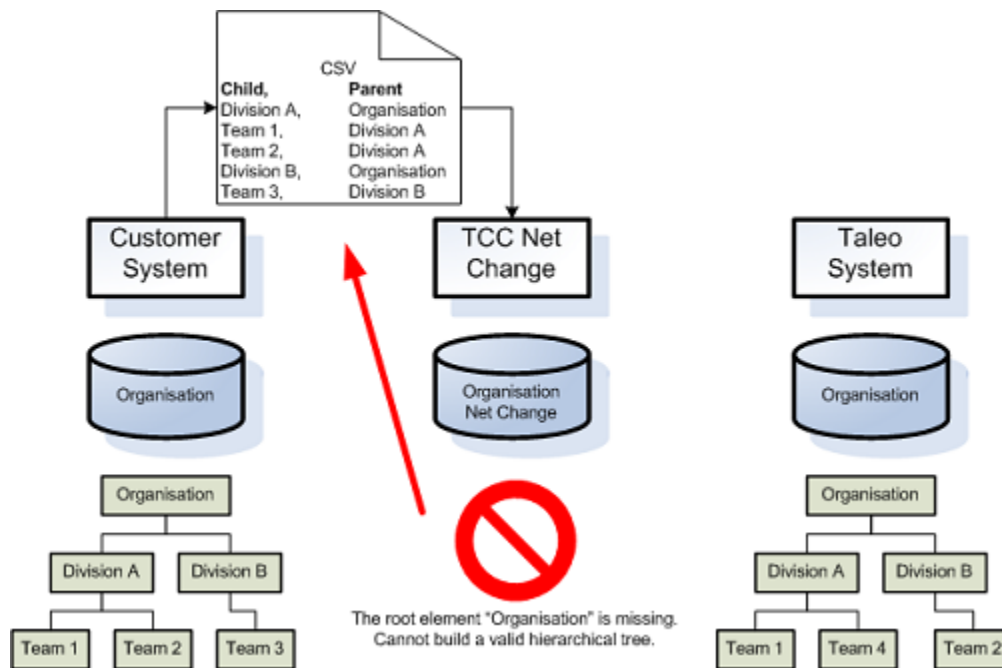


## Final Result

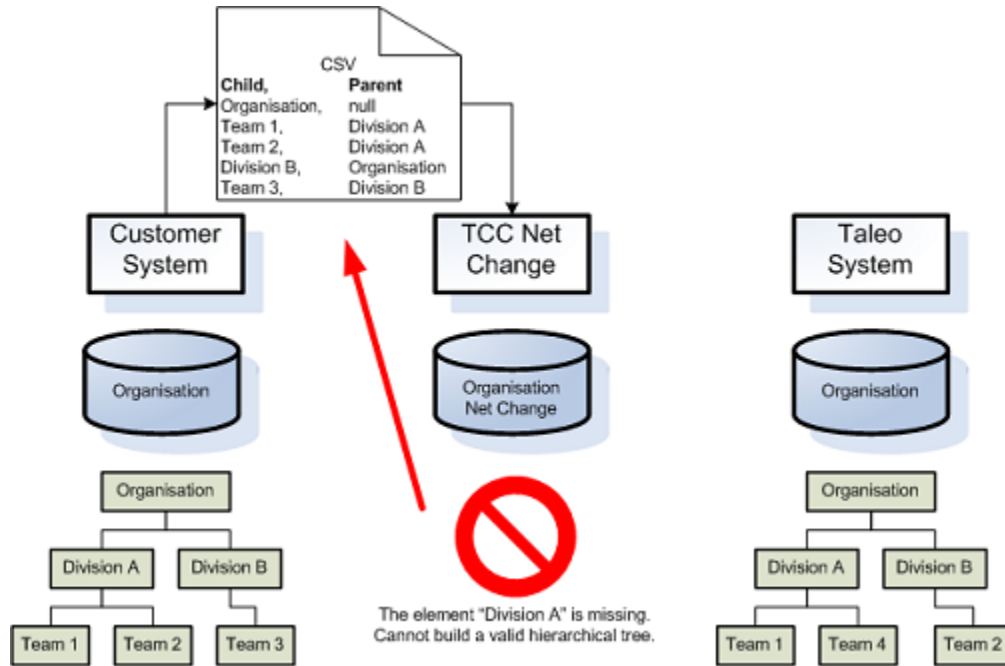


## Invalid Requests

### Missing Root

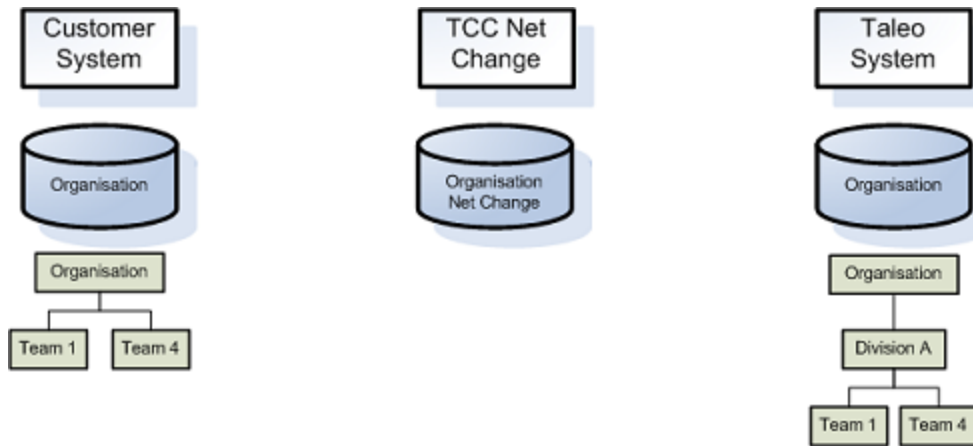


### Missing Element



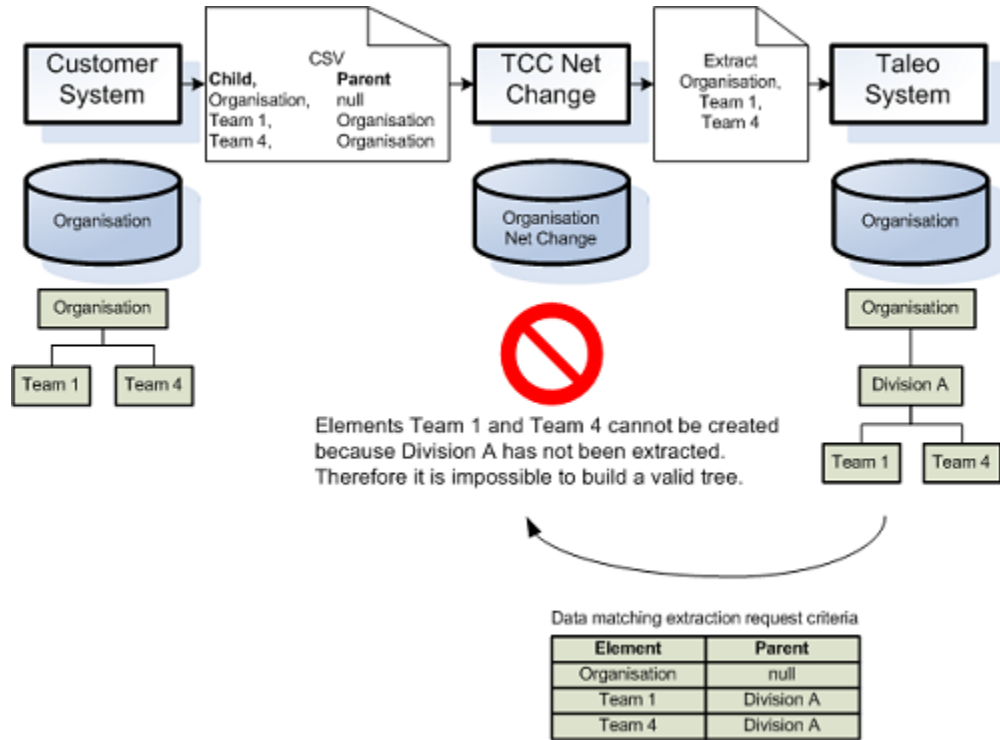
## Different Structures

### Prior to Net Change Setup



The Net Change Setup of the Net Change database is performed with the keys provided in the Customer input file. The Net Change Setup will fail to create a valid hierarchical structure if the Customer and Taleo structures do not match.

### Net Change Setup Data Extraction



## User Entity Constraints

During User Net Change Setup, it is impossible to extract the values for the Password and ForceChangePassword parameters. Therefore, the Net change database is not perfectly synchronized with the Taleo zone.

The User Password and ForceChangePassword parameters require special treatment during first Net Change. Protecting or overwriting the values for those parameters in the Taleo zone determines the treatment to apply.

### Protect Values

To protect a password that has already been modified by the user, specify the columns corresponding to the Password and ForceChangePassword parameters as non-updatable. In the advanced options section, enter updatable for the name and false for the value.

### Overwrite Values

To overwrite the Password and ForceChangePassword parameter values in the Taleo zone, set the updatable value to true in the advanced options section. This is the default value for the option, therefore it does not need to be specified. During Net change first run all values will be overwritten and the databases will be synchronized.

## Setup Usage

The Net Change Setup feature must be used in the implementation phase to fill the Net Change database with target application data.

It can also be used to resynchronize the Net Change database with the target application data. This should be done only in exceptional cases.

It must also be used when the entity key changes. The Net Change tables need to be recreated.

Note that only the Net Change tables for the entity being processed will be deleted at the start of the Net Change Setup process.

Also note that no backup is made before erasing the tables.

## Known Limitations

The following limitations must be taken into account when implementing a solution that uses Net Change.

### Entity Delete Mode Selection

The delete mode is currently defined in the entity Net Change definition file (feature pack). Only one delete mode is defined. The actual solution does not support the selection of different delete modes per configuration.

### Attachment File Support

Attachment files are not supported by Net Change.

### Entity Key Modification

If you change a defined entity key after the first Net Change run, the Net Change database for this entity must be recreated. Use the Net Change Setup feature to recreate Net Change database.

### storage.properties File Modification

The storage.properties file parameters determine how Net Change stores data in its reference database. The properties that can be set are data compression and encryption, repository location, storage unit interface, and block size.

The values defined in the storage.properties file (`UseCompression`, `EncryptionMode`, `StorageUnitImplementation`, and `FileStorageUnit.DefaultBlockSize`) must not be changed after the first Net Change run.

If this is required, **ALL** Net Change database tables must be recreated. Perform a Net Change Setup on every entity subject to a Net Change process.

### Multiple Database Synchronization

Net Change maintains a single database to synchronize a customer system data and a Taleo Product data. This implies installing and running a separate instance of TCC for each customer system synchronizing with a Taleo product. For example, synchronizing three different Human Resource Information System (HRIS) databases with one Taleo zone means installing and running three separate instances on TCC. The data provided for synchronization by each HRIS must be mutually exclusive to prevent reversal of operations. For example, if one system deletes an entity on

an ongoing integration and the same entity exists on another system, it will be recreated when that system runs its integration.

The same principle applies to synchronizing customer HRIS data with several Taleo zones. For example, synchronizing one customer HRIS with a Taleo production zone and a Taleo staging zone requires installing and running two separate instances of TCC.

## Net Change Configuration

The TCC Net Change plug-in uses a configuration file to define properties related to its database. The file name is `storage.properties`

### Properties

- **UseCompression:**

Determines the compression or not of stored Net Change data. This property is set to true in the default configuration file, the data is compressed. If the property is removed from the configuration file, the value will default to false, the data will not be compressed.

- **EncryptionMode:**

Determines the type of encryption to apply on Net Change data. This property is set to 2 in the default configuration file.

0=No encryption

1=Custom TCC internal encryption

2=AES 128-bit ciphers

- **RepositoryLocation:**

Determines Net Change data files repository location. This property is empty in the default configuration file. The default location is `[TCCInstallFolder]/system/repository`. If the property is absent or empty the default location is used.

- **StorageUnitImplementation:**

Advanced property, modify only as instructed by Taleo.

Determines Storage Unit interface implementation. The default value is `com.taleo.integration.storage.FileStorageUnit` and is the only value supported at this time.

- **FileStorageUnit.DefaultBlockSize:**

Advanced property, modify only as instructed by Taleo.

Determines file storage block size in KB (min. 1, max. 128). This property is set to 1 in the default configuration file. If the value is not specified or invalid, the value will default to 2.

### Overriding a plug-in configuration location

There are two ways to specify the location of plug-in configuration files:

**Default:**

- By default, the plug-in configuration file location is:  
[PLUGIN\_FOLDER]\[PLUGIN\_COLLECTION\_NAME]\configuration.
- Nothing special needs to be setup.
- E.g.: The file “storage.properties” will be located at “c:\Program Files\Taleo Connect Client\extensions\plugins\tcc-netchange\storage.properties”

**Root configuration folder:**

- To change the location of plug-in configuration file for every plug-in at the same time.
- System Property: “com.taleo.integration.client.extensions.plugins.configuration.dir”.
- E.g: com.taleo.integration.client.extensions.plugins.configuration.dir=c:\PluginConfig.The file “storage.properties” will be located at: c:\PluginConfig\tcc-netchange\storage.properties.



# 7 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 CieInstitution Number.

Must exist in Taleo CieInstitution 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.

- **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) Loginname.

- **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**  
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.

- **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.

- **Justification**

1=New Position, 2=Replacement, 3=Modification, 4=Extensio.

- **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 Groups1=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.



# 8 Advanced Queries

## 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

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.

1. Click on Projections tab.
  - a. Click on Add.
  - b. Select Add a complex projection.
  - c. Click on OK.

The Complex projection pane displays: `<quer:projection xmlns:quer="http://www.taleo.com/ws/integration/query"/>`

2. Enter or paste code in Complex projection pane.

```
<quer:projection alias="Candidate_Anonymous" xmlns:quer="http://www.taleo.com/ws/integration/query">
  <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>
  </switchByCriterion>
</projection>
```



```
</case>
</cases>
<defaultValue>
<string>Unknown candidate</string>
</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
'Unknow 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>
    <projection projectedField="firstName">
      <concatenate>
        <concatenate>
          <field path="firstName"/>
          <string> </string>
        </concatenate>
        <field path="lastName"/>
      </concatenate>
    </projection>
  </projections>
</query>
```

## 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">
      <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

```
SELECT DBMS_LOB.SUBSTR(CSUSERATTACHEDFILE01.FILECONTENT, 3, 4) AS blobSubstring
FROM CSUSERATTACHEDFILE CSUSERATTACHEDFILE01
WHERE (CSUSERATTACHEDFILE01.CSUSERATTACHEDFILENO=10291)
```

### XML query API

```
<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

```
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>
<field path="firstName"/>
</decode>
</projection>
</projections>
</query>
```

## Distinct

### SQL

```
select distinct currency from csuser
```

### XML query API

```
<query alias="subQuery" preventDuplicates="true" projectedClass="com.taleo.akirademo.entity.Candidate">
<projections>
<projection alias="currency">
<field path="currency"/>
</projection>
</projections>
</query>
```

## Dual

### 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>
```

## 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">
            <projections>
              <projection>
                <field path="salary"/>
              </projection>
            </projections>
          </query>
        </returnsResults>
      </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">  
  <projections>  
    <projection alias="candidate">  
      <field path="candidate"/>  
    </projection>  
    <projection alias="appCount">  
      <count>  
        <field path="candidate"/>  
      </count>  
    </projection>  
  </projections>  
  <groupings>  
    <grouping>  
      <field path="candidate"/>  
    </grouping>  
  </groupings>  
</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)
```

### XML query API

```
<query alias="testGroupingFilteringInReportQuery" projectedClass="com.taleo.akirademo.entity.Candidate">  
  <projections>  
    <projection>  
      <field path="city"/>  
    </projection>  
    <projection alias="cityCount">  
      <count>  
        <field path="city"/>  
      </count>  
    </projection>  
  </projections>  
</query>
```



```
</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>
```

## In 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 '\'))))
```

```
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE (CSUSER01.CSUSERNO IN (1, 2, 3))
```

## XML query API

```
<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>

<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"/>
      </includedIn>
    </filtering>
  </filterings>
  <list>
    <integer>1</integer>
    <integer>2</integer>
    <integer>3</integer>
  </list>
</query>
```

## Inner Join

### 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

```
<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

```
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

```
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE (CSUSER01.FIRSTNAME LIKE 's%' ESCAPE '\')
```

### XML query API

```
<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>
```

```
</query>
```

## Minus

### 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

```
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE ((CSUSER01.FIRSTNAME LIKE 'a%' ESCAPE '\') OR
(CSUSER01.FIRSTNAME='Sam'))
```

### XML query API

```
<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>
```

```
</filtering>  
</filterings>  
</query>
```

## Order By

### SQL

```
select csuserno from csuser order by csuserno asc
```

```
select csuserno from csuser order by csuserno desc
```

### XML query API

```
<query alias="testOrderByAsc" projectedClass="com.taleo.akirademo.entity.Candidate">  
  <projections>  
    <projection>  
      <field path="no" />  
    </projection>  
  </projections>  
  <sortings>  
    <sorting ascending="true">  
      <field path="no" />  
    </sorting>  
  </sortings>  
</query>
```

```
<query alias="testOrderByDesc" projectedClass="com.taleo.akirademo.entity.Candidate">  
  <projections>  
    <projection>  
      <field path="no" />  
    </projection>  
  </projections>  
  <sortings>  
    <sorting ascending="false">  
      <field path="no" />  
    </sorting>  
  </sortings>  
</query>
```

## 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  
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">
  <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>  
      <truncate>  
        <field path="salary"/>  
      </truncate>  
    </projection>  
  </projections>  
</query>
```

## Union

### 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

```
<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=""/>
        </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>
</unionAllowDuplicates>
```

## Other Custom Functions

### SQL

```
SELECT CUSTOM_FUNCTION_NAME(myField,'param1','param2') AS myField01
FROM CSUSER CSUSER01
WHERE 1=1
```

### XML query API

```
<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

```
SELECT REGEXP_REPLACE(CSUSER01.USERNAME,?,?) AS USERNAME01
FROM CSUSER CSUSER01
```

WHERE 1=1

## XML query API

```
<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 cannot be used as an Alias for an export query. The following table lists words that are reserved in Oracle.

| Oracle Reserved Words |                |            |           |
|-----------------------|----------------|------------|-----------|
| ABORT                 | ACCEPT         | ACCESS     | ADD       |
| ALL                   | ALTER          | AND        | ANY       |
| ARRAY                 | ARRAYLEN       | AS         | ASC       |
| ASSERT                | ASSIGN         | AT         | AUDIT     |
| AUTHORIZATION         | AVG            | BASE_TABLE | >BEGIN    |
| BETWEEN               | BINARY_INTEGER | BODY       | BOOLEAN   |
| BY                    | CASE           | CHAR       | CHAR_BASE |
| CHECK                 | CLOSE          | CLUSTER    | CLUSTERS  |
| COLAUTH               | COLUMN         | COMMENT    | COMMIT    |
| COMPRESS              | CONNECT        | CONSTANT   | CRASH     |
| CREATE                | CURRENT        | CURRVAL    | CURSOR    |
| DATA_BASE             | DATABASE       | DATE       | DBA       |

| Oracle Reserved Words |            |            |                |
|-----------------------|------------|------------|----------------|
| DEBUGOFF              | DEBUGON    | DECIMAL    | DECLARE        |
| DEFAULT               | DEFINITION | DELAY      | DELETE         |
| DESC                  | DIGITS     | DISPOSE    | DISTINCT       |
| DO                    | DROP       | ELSE       | ELSIF          |
| END                   | ENTRY      | EXCEPTION  | EXCEPTION_INIT |
| EXCLUSIVE             | EXISTS     | EXIT       | FALSE          |
| FETCH                 | FILE       | FLOAT      | FOR            |
| FORM                  | FROM       | FUNCTION   | GENERIC        |
| GOTO                  | GRANT      | GROUP      | HAVING         |
| IDENTIFIED            | IF         | IMMEDIATE  | IN             |
| INCREMENT             | INDEX      | INDEXES    | INDICATOR      |
| INITIAL               | INSERT     | INTEGER    | INTERFACE      |
| INTERSECT             | INTO       | IS         | LEVEL          |
| LIKE                  | LIMITED    | LOCK       | LONG           |
| LOOP                  | MAX        | MAXEXTENTS | MIN            |
| MINUS                 | MLSLABEL   | MOD        | MODE           |
| MODIFY                | NATURAL    | NATURALN   | NETWORK        |
| NEW                   | NEXTVAL    | NOAUDIT    | NOCOMPRESS     |
| NOT                   | NOWAIT     | NULL       | NUMBER         |
| NUMBER_BASE           | OF         | OFFLINE    | ON             |
| ONLINE                | OPEN       | OPTION     | OR             |

| Oracle Reserved Words |           |             |           |
|-----------------------|-----------|-------------|-----------|
| ORDER                 | OTHERS    | OUT         | PACKAGE   |
| PARTITION             | PCTFREE   | PLS_INTEGER | POSITIVE  |
| POSITIVEN             | PRAGMA    | PRIOR       | PRIVATE   |
| PRIVILEGES            | PROCEDURE | PUBLIC      | RAISE     |
| RANGE                 | RAW       | REAL        | RECORD    |
| REF                   | RELEASE   | REMR        | RENAME    |
| RESOURCE              | RETURN    | REVERSE     | REVOKE    |
| ROLLBACK              | ROW       | ROWID       | ROWLABEL  |
| ROWNUM                | ROWS      | ROWTYPE     | RUN       |
| SAVEPOINT             | SCHEMA    | SELECT      | SEPERATE  |
| SESSION               | SET       | SHARE       | SIGNTYPE  |
| SIZE                  | SMALLINT  | SPACE       | SQL       |
| SQLCODE               | SQLERRM   | START       | STATEMENT |
| STDDEV                | SUBTYPE   | SUCCESSFUL  | SUM       |
| SYNONYM               | SYSDATE   | TABAUTH     | TABLE     |
| TABLES                | TASK      | TERMINATE   | THEN      |
| TO                    | TRIGGER   | TRUE        | TYPE      |
| UID                   | UNION     | UNIQUE      | UPDATE    |
| USE                   | USER      | VALIDATE    | VALUES    |
| VARCHAR               | VARCHAR2  | VARIANCE    | VIEW      |
| VIEWS                 | WHEN      | WHENEVER    | WHERE     |

| Oracle Reserved Words |      |      |       |
|-----------------------|------|------|-------|
|                       |      |      |       |
| WHILE                 | WITH | WORK | WRITE |
| XOR                   |      |      |       |



# 9 Security

## 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. TCC supports TLS 1.2.

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.

## User Authentication

TCC supports basic user name and password authentication as well as certificate based authentication.

### Web Services Security (WS-S)

Oracle Taleo announced the deprecation of support for Username Token Profile 1.1 using the PasswordDigest scheme in release 14B. Any customers using this method for authenticating with the Oracle Taleo Enterprise Edition 14B and above zones, will be required to make modifications and replace Password Digest with Basic Authentication. Configuration that were using this scheme will now automatically default to basic authentication.

### Basic Authentication

When using the basic authentication, TCC provides user credentials in the form of a user name and password, as HTTP headers.

### Certificate Based Authentication

TCC supports the signing of communication between TCC and the Taleo zone. The authentication is used to sign the requests and responses between Connect Client and Taleo. This is in addition to basic authentication.

## Stored Passwords

Taleo Connect Client (TCC) communicates to Taleo hosted products using an authenticated user and password. In addition, if proxy servers are installed within the environment additional user credentials can be required by TCC.

## Integration User password

TCC maintains the Integration User password locally to establish a connection with a Taleo hosted product. Passwords can be stored in the following locations:

- Default configuration board file: default.board.xml
- Environment file: \*.properties
- 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:

```
EncryptedPassword=Base64 (AES:128 (password))
```

## Encryption Key Server (EKS)

With release 14A, an encryption key server is delivered with Taleo Connect Client (TCC) which can be used to augment existing password management services. The EKS can be used to generate a key that encrypts passwords within the TCC application.

## Encryption Key Server Installation

Step 1. Copy the encryptionkey-server.jar file from the server folder under the TCC installation folder, to a location on the network. It is recommended that this be a different location than the TCC installation. The encryptionkey-server.jar file is the default EKS delivered for enhancing password management.

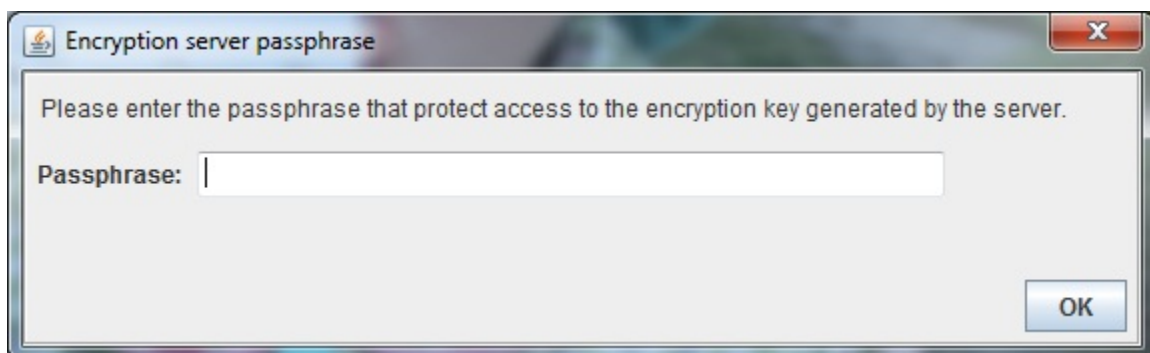
Step 2. By default, the EKS runs using a local host IP address and port 1126. To use another address or port, a folder and file must be created at the server root location. (A server restart is required following any modification.)

Folder and file name:

- conf/EncryptionKeyServer.conf File content:
- HOST\_PORT=YOUR\_SERVER\_PORT
- HOST\_ADDRESS=YOUR\_SERVER\_ADDRESS

Step 3. The encryption server can now be started, call your\_java\_location -jar encryptionkey-server.jar.

a. On Windows platform - the following screen prompts is displayed to the user. The user should enter a passphrase value. This value is masked with \*.



b. On platform - command line will prompt user for passphrase entry.

On starting the EKS for the first time, an AES symmetric encryption key is generated. This will be automatically saved into the conf/ClientKey.key file. The user entered passphrase is used to protect access to the encryption key value stored in this file. Each new installation of the EKS will generate a different & unique encryption key. Due to this, it is strongly recommended that a backup of the key file is made and saved in a secured location. In doing so, it is possible to recover encrypted data if there is a failure with the original implementation. Using the same passphrase would unlock the encryption key needed to decrypt the data. At all subsequent EKS startups, the entered passphrase will be used to unlock the encryption key needed to decrypt any encrypted data.

## Configuring TCC using an Encryption Key Server

It is necessary for TCC to be provided with the host address and port of the EKS that was setup and configured in the previous step. This can be added when a new instance of TCC is being installed via the setup dialog step illustrated below. This information is captured during the installation process of the new version of TCC and stored in the TaleoConnectClient.ini file.

If more convenient, the EKS can be installed at a later time, and the relevant host and port information can be added directly into the TaleoConnectClient.ini file as illustrated below:

- -Dcom.taleo.integration.client.encryption.key.hostPort=*Your\_Server\_Port*
- -Dcom.taleo.integration.client.encryption.key.hostAddress=*Your\_Server\_Address*

Please note that using a different encryption server will imply modifying the configuration files to use the new encryption key.

In TCC, no modifications are visible in the UI. When creating a new profile or configuration board entry, if both properties have been specified, the encryption from the server is used to encrypt the password. Otherwise, the default encryption mechanism is used.

## Encrypt Password Tool

An additional tool is provided to encrypt individual string values. It is located in the **Tools > Encrypt Password** menu.

Enter the string value to encrypt in the text box and click *Encrypt*. If the encryption server properties are defined, this will be used to encrypt the value. Otherwise, the default encryption mechanism will be used. The encrypted result is displayed at the bottom of the page. Click *Copy* to save to clipboard. Click *Close* to close the window.

## Proxy Password Management Using EKS

TCC uses a System Properties file to authenticate itself against proxy servers needed to access the internet. The required System properties are:

```
-Dhttp.proxyHost=myproxyhostname
```

```
-Dhttp.proxyPort=portnumber
```

```
-Dhttp.proxyUser=username
```

```
-Dhttp.proxyPassword=hello123
```

```
-Dhttps.proxyPassword=securedHello123
```

**Note:** Values defined for properties `http.proxyPassword` and `https.proxyPassword` need to be encrypted. The new Encryption Password Tool can be used to encrypt different String values. The presence of the Encryption Key Server address and port properties is used to determine the encryption mode.

The two following properties must be added to use the Encryption Key Server:

```
-Dcom.taleo.integration.client.encryption.key.hostAddress=your_server_address
```

```
-Dcom.taleo.integration.client.encryption.key.hostPort=your_server_port
```

There are two different ways to set up System properties, depending on the way TCC is launched.

1. On windows via the GUI:

When running TCC through the Windows GUI, all System properties must be defined in the `TaleoConnectClient.ini` file located at the root of the installation folder.

2. Launching TCC through the 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` (Linux) files.

To be able to run TCC from command line without any issue it is recommended to use the following Java options (don't forget to replace the variables with the values that are best for the type of operations that are done).

```
JAVA_OPTS="
-ms${JAVA_MIN_MEM}M
-mx${JAVA_MAX_MEM}M
-XX:NewSize=${NEWGEN}m
-XX:MaxNewSize=${MAXNEWGEN}m
-XX:PermSize=${PERMGEN}m
-XX:MaxPermSize=${MAXPERMGEN}m
```

## EKS Logging Service

A logging service is available for operations related to the encryption key server. Logging is written in file `log/EncryptionServerLog.log`. Default level is INFO. It is possible to change the logging level using a property at startup. Here are the possibilities:

- `-Dlog.file.level=LEVEL_TO_USE`
- `-Djava.util.logging.config.file=YOUR_LOGGING_PROPERTIES_FILE_PATH`

Here is an example of logging properties file content: `com.taleo.integration.client.encryption.key.level=FINE`

## 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.

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:
xH/JD4X3xuVTMz7pO1fdxcFUBULFWGrZqTfNDw/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: [DECRYPT("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/JD4X3xuVTMz7pO1fdxcFUBULFWGrZqTfNDw/DLpo=\")]
```

## 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://docs.oracle.com/javase/1.5.0/docs/guide/net/properties.html> (these are the same specifications as for the 1.6 <http://docs.oracle.com/javase/6/docs/technotes/guides/net/properties.html> or 1.7 <http://docs.oracle.com/javase/7/docs/technotes/guides/net/properties.html> 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

```
%JAVA_HOME%\bin\java.exe
-Dhttp.proxyHost=proxy.acme.net
-Dhttp.proxyPort=3128
-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:

```
%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:

```
vmargs
-Djava.endorsed.dirs=lib/endorsed
-Dorg.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
-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

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 Support

TCC supports both Microsoft's NTLM V1 and V2 proxy server authentication schemes. 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 proxy server authentication scheme must be configured in files located at the root of the TCC installation folder.

- **Launching TCC using 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.

## Data at Rest

Standard TCC implementations have no data retention capabilities. Data-at-rest encryption capabilities are not required.



# 10 Appendix

## Appendix A

### Volume Limits

The volume limits specifies maximums for the number of records per day or transaction.

#### Basic Offering

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

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

#### Concurrent Integrations and Frequency

It is possible to run parallel Taleo Connect integrations against a customer's zone. Currently this is set to 4 Concurrent Integrations per Taleo Product. So it is possible to run 4 imports at a time, or 3 import and 1 export at a time, etc, or 4 exports at a time per Taleo Product. All subsequent requests will be queued and run later when there is an available slot to run them.

Any one Taleo Connect integration can be scheduled up to 15 minutes intervals to preserve the zone's level of service to the users.

#### 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.

| Project Type                              | Daily Import | Daily Export | Per Transaction Export  |
|---|--------------|--------------|-------------------------|
| Data Migration (New Customer)             | 350 000      | 500 000      | 100 000 records, 256 MB |
| Data warehouse initial extract            | N/A          | 10 million   | 100 000 records, 256 MB |
| Final Extract                             | N/A          | 10 million   | 100 000 records, 256 MB |
| Large scale data refresh or modifications | 350 000      | 5 million    | 100 000 records, 256 MB |

## 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.

Customers are only permitted a maximum of 3 planned exceptions per calendar year.

## 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 is 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

Taleo Connect Client compatibility with Taleo products is guaranteed one version up, when the same reference model is used. When the reference model used to create an integration point changes, compatibility is not guaranteed.

## 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.

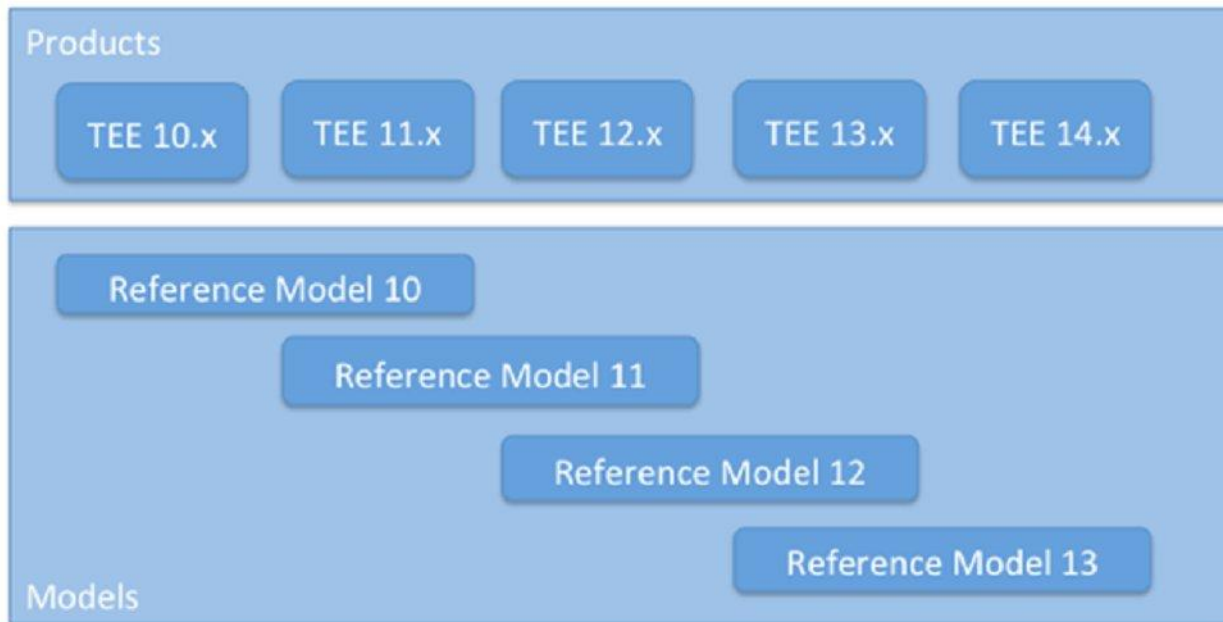
For information on the integration reference model changes for each release, refer to the following Notes.

- See: Taleo Enterprise - Taleo Integration Release Notes (MOS under Release Notes, Maintenance Packs, Express Packs, and Incidents Resolved History)

## Example

An integration point built with reference model 11 X for a version 11.X product will continue to function after the product is upgraded to version 12 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:** Integrations built using the 7.5 reference model are expected to cease to be formally supported mid to late 2014, though formal communication will be made 12 months in advance of full compatibility support stopping. It is recommended in preparation to upgrade integration scripts currently on 7.5 to the latest data model version appropriate for your zone's version or at least begin the planning activities associated with this upgrade, subsequent testing and deployment.

The Recruiting/Professional and SmartOrg models have changed considerably from 7.5 upwards and it is recommended referencing the following guides for further information.

- See: Taleo Integration Migration Guide (MOS under Release Notes, Maintenance Packs, Express Packs, and Incidents Resolved History)
- See: Product Data dictionaries (MOS under Release Notes, Maintenance Packs, Express Packs, and Incidents Resolved History)

## Migration

Given the dependence on the Taleo zone data model for successful integrations, it is important to maintain compatibility by migrating and upgrading integrations to newer data model versions.

This may require an upgrade to new versions of TCC or simply the application of newer versions of PIP files to an existing instance of TCC.

### Migration Process

The following process should be followed to upgrade an integration script.

1. Download and install the latest version of the product PIP file appropriate for your zone..
2. Install the PIP file in the default location for PIP for your TCC instance.
3. Open TCC and select the migration helper tool.
4. Using the migration helper tool, select the import or export files to upgrade from the existing model to the new model corresponding to the PIP downloaded in step 1.
5. The output from the migration helper tool is a new file named with the extension `*_migration_*.xml`. Modify the extension and location to correspondence to your existing deployment package requirements and setup.

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.