Clay Tablet™ TECHNOLOGIES

Clay Tablet Connector for Sitecore

Installation and Configuration Guide

Version 3.9.11

February 8, 2017

Copyright

Although the information in this document has been carefully reviewed, Clay Tablet does not warrant it to be free of errors or omissions. Clay Tablet reserves the right to make corrections, updates, revisions, or changes to the information in this document.

Contents

1 Welcome to the Clay Tablet Connector for Sitecore

Welcome to the Clay Tablet Connector for Sitecore ("Connector"). This is Clay Tablet's connector between Sitecore and the Clay Tablet Platform.

1.1 Terminology

Amazon AWS	Amazon Web Services. A suite of web application products developed and sold by Amazon.com. Clay Tablet uses various AWS offerings in order to leverage their infrastructure and build rich, dynamic solutions for its customers, specifically, the Clay Tablet Platform. For details, see http://aws.amazon.com.
Amazon S3	Amazon Simple Storage Service. For details, see: http://aws.amazon.com/s3/. The Connector and the Clay Tablet Platform use Amazon S3 to provide temporary storage services for the content sent to and from translation.
Amazon SQS	Amazon Simple Queue Service. For details, see: http://aws.amazon.com/sqs/. The Connector uses Amazon SQS to provide Message Queue Services.
Asset	Any content or document being sent for translation, including metadata. Assets are created by the Connector.
Clay Tablet (CTT)	Clay Tablet Technologies, the corporate entity that publishes the Clay Tablet Connector and the Clay Tablet Platform.
Clay Tablet Connector for Sitecore ("Connector")	The connector software that Clay Tablet Technologies has developed and provides, which plugs into your Sitecore CMS to provide connectivity to our hosted Platform. In this document it is referred to as the Connector. This is the software you are installing and configuring as you work through this document.
Clay Tablet Platform	The hosted (IaaS) connectivity platform that receives and routes content from content systems, including content management systems (CMSs), to translation providers and back during implementation. Clay Tablet Technologies configures the Platform based on the number and nature of systems involved in your system architecture.
CT3	Legacy branding term that refers to the Connector.
FTP Server	File Transfer Protocol (FTP) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet. Translation providers may receive and send files for translation using an FTP server.
IaaS	Infrastructure as a Service. The Clay Tablet Platform is an IaaS, because it is a hosted platform.

Keys	The Connector uses keys to establish a secure, discrete connection between the Connector instance and the Platform. **Very important:** Do not copy the CMS address keys to multiple Sitecore instances, because this is a violation of the Clay Tablet License Agreement. Using the same CMS address keys on multiple Sitecore instances will cause the Connector to behave unexpectedly, which can result in lost translation content, orphaned projects, and inaccurate translation status reports. Clay Tablet will only support technical issues caused by duplicating or incorrectly installing CMS address keys on a time and materials basis.
MT	Machine translation. The translation provider can be a machine translation service, such as Google Translate.
On-Premise Platform	A version of the Clay Tablet Platform that is hosted and managed by the Clay Tablet client, instead of hosted on AWS by Clay Tablet.
Producer	CMS or another system that sends content or documents out for translation. In this case, this is your Sitecore Content Editor.
Provider	A provider of translation services. The delivery of assets to the provider may be via an FTP server or a TMS connector.
Support Asset	Supporting documents and their metadata. Support assets are not translated by the translation provider, but they provide helpful context for the translator.
TMS	Translation management system that the translation provider users.

1.2 About the Clay Tablet Translation Platform

Clay Tablet’s translation connectivity platform is the easiest, most flexible way to integrate content systems, including content management systems (CMSs) and other content producers, with translation providers and translation technologies.

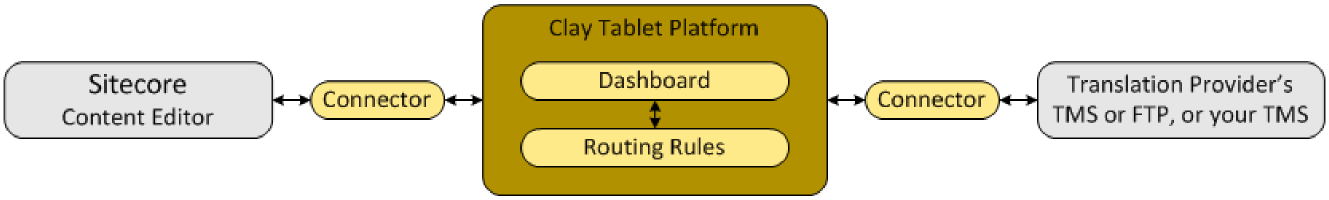
Clay Tablet Platform is the hosted (IaaS) connectivity platform that receives and routes content from content management systems to translation providers and back. It is hosted on Amazon Web Services (AWS). During implementation, Clay Tablet Technologies configures the Platform for your translation solution, based on the translation providers or systems you use. The Clay Tablet Platform uses the following services on AWS:

- S3 (Amazon Simple Storage Service), which provides storage services for the content sent to and from translation.
- SQS (Amazon Simple Queue Service), which provides message queue services.

1.3 How the Clay Tablet Connector Works with Sitecore

The Clay Tablet Connector ("Connector") is an important part of the Clay Tablet translation solution.

The Connector is installed on your system as an add-in to the Sitecore CMS. Its functionality is displayed to the users as part of the Sitecore Content Editor, Page Editor (Sitecore 6 and 7), Experience Editore (Sitecore 8) and Desktop.



Your translation systems architecture might look like the configuration above. It may have additional CMSs or translation providers, but the core concepts remain the same.

During implementation, Clay Tablet works with you and your translation providers to configure and test the other elements of your translation solution, which are the Clay Tablet Platform's connections to your translation providers' systems.

1.4 Using this Guide

Purpose of this guide

This guide describes everything you need to know to install and configure the Clay Tablet Connector ("Connector") for Sitecore. It describes the delivery package contents, system requirements, installation instructions, and configuration procedures.

Recommendation: Review the user guide to fully understand the powerful features of the Connector.

Who should use this guide

This guide is intended for Sitecore administrators and system integrators.

What you should already know

This document assumes that your company already has an installed instance of Sitecore. It assumes that you have a strong working knowledge of the Sitecore Content Editor and Sitecore features, specifically how to configure workflows and associate them with templates.

How to find out more about the Clay Tablet Connector for Sitecore

For information on using the Clay Tablet Connector to send and receive content for translation from Sitecore, read the *Clay Tablet Connector for Sitecore User Guide*.

Documentation conventions

This guide uses the following conventions:

Convention	Description
Bold	Highlights screen elements such as buttons, menu items, and fields.
`Courier`	Highlights input, file names, and paths.
Italics	Highlights terms to emphasize, variables, or document titles.
>	Indicates a menu choice. For example, "Select **Sitecore Desktop > All Applications > CTC Translation > Translation Filter**."

1.5 How to Contact Clay Tablet Support

Email @: support@clay-tablet.com

Telephone: +1-416-363-0888

You can submit a support ticket either:

- by email
- from the Clay Tablet Zendesk page, using your web browser

To submit a support ticket:

1. Do one of the following:
   - Email support@clay-tablet.com, and cc (carbon copy) anyone to include in the ticket correspondence.

     Important: Include the information and attachments in your email that are listed in the sub-sections below.
   - Create a ticket in Zendesk:

     a. Open the Clay Tablet Zendesk page in your browser: https://claytablet.zendesk.com.

     b. Sign in to Zendesk. If you do not have sign-in credentials, see "To view and update your support ticket in Zendesk:" below.

     Important: Include the information and attachments that are listed in the sub-sections below.

     c. Click **Submit a request**.

     d. In the **CCs** field, add anyone to include in the ticket correspondence.

   Zendesk automatically creates a ticket and responds to everyone included in the cc field.

2. Everyone in the original cc receives updates unless they request to be removed from the ticket.

Important: Check your email spam folder (especially first-time Zendesk users) as sometimes email notifications from Zendesk are marked as spam.

When the issue is resolved, Clay Tablet closes the ticket.

Information to include in the support ticket:

- client name
- CMS or content system name and version
- Connector or App version installed
- name of job for which the issue occurs
- date of job submission
- detailed description of the issue
- any error text–copy and paste, if applicable

Files to attach to the support ticket:

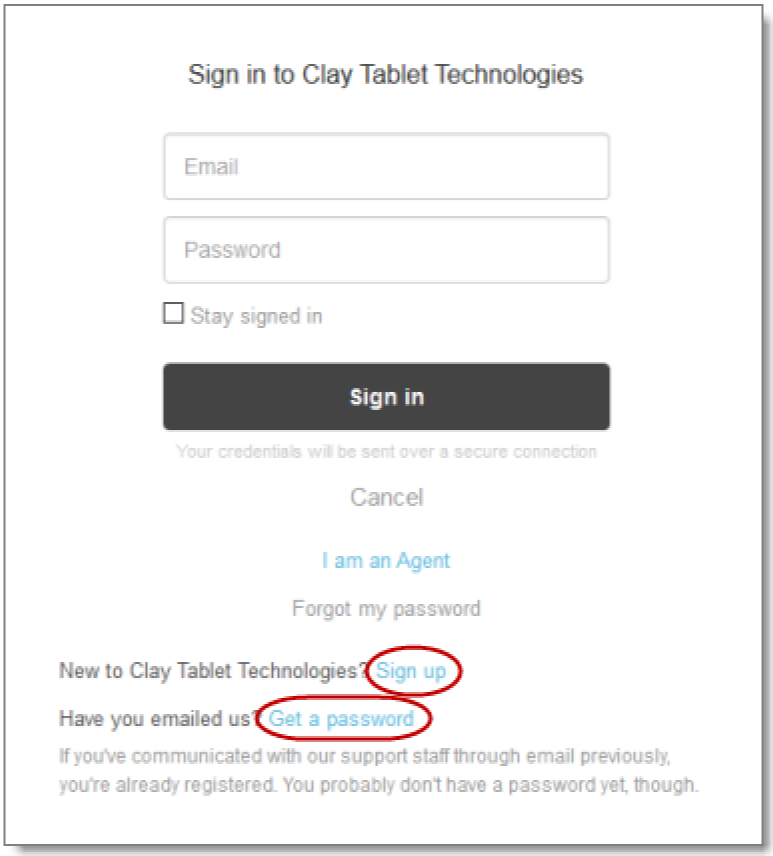
- CMS log files for the date the issue occurred
- Clay Tablet log files for the date the issue occurred
- screen capture of the issue

To view and update your support ticket in Zendesk:

Important: You must log into Zendesk to view your support tickets there.

1. Open the Clay Tablet Zendesk page in your browser: https://claytablet.zendesk.com.
2. In the top-right corner, click **Sign in**, and enter your credentials.

   Note: If you do not have sign-in credentials yet, then click either **Sign up** or **Get a password**, and follow the onscreen instructions.



Sign in to Clay Tablet Technologies

Email

Password

□ Stay signed in

Sign in

Your credentials will be sent over a secure connection

Cancel

I am an Agent

Forgot my password

New to Clay Tablet Technologies? Sign up

Have you emailed us? Get a password

If you've communicated with our support staff through email previously, you're already registered. You probably don't have a password yet, though.

3. After signing in, click **My activities** to view the tickets you opened or where you are cc'd.

4. To update tickets, you can reply or attach files.

For more information, refer to "Submitting and tracking support requests" in Zendesk's *Help Center guide for end-users*, at: https://support.zendesk.com/hc/en-us/articles/203664386-Help-Center-guide-for-agents-and-end-users.

Important: Zendesk refers to a *support ticket* as a *support request*. These terms are interchangeable.

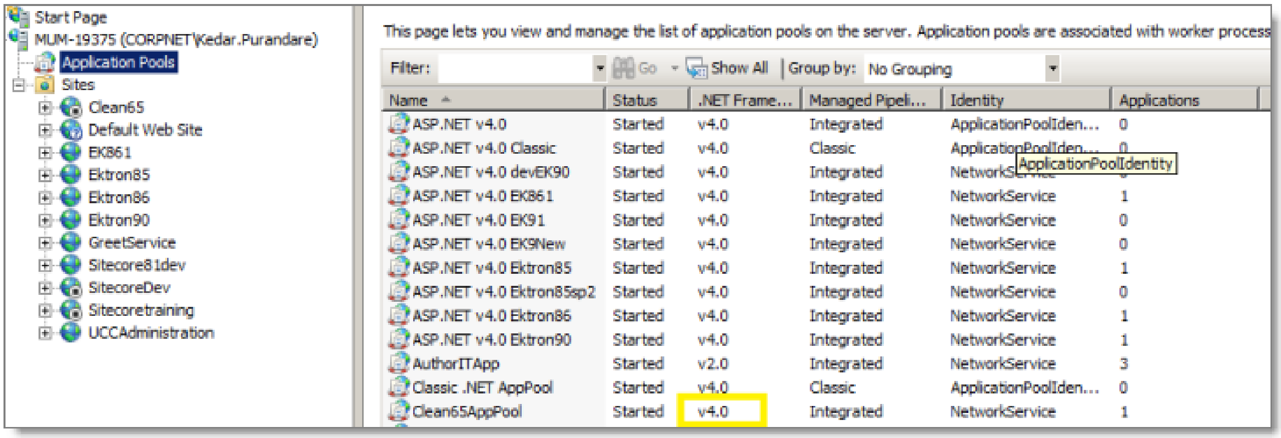
2 Before You Install

Before you begin to install the Clay Tablet Connector for Sitecore, please review the system requirements, described below, and perform the following pre-installation procedures:

1. "Setting Your System Date, Time, and Time Zone Correctly" on page 13.
2. "Downloading the Delivery Package" as described on page 13. This describes the contents of the Connector delivery package that Clay Tablet Technologies sends you.
3. "Setting Up the Translation Database" as described on page 14. This section includes information about database size requirements as well as detailed instructions for Oracle and Microsoft SQL Server databases.
4. "Creating the Connector Folders" as described on page 18. These are the folders where the Connector stores your licensing information and your translation data.
5. "Installing the CMS Address Key and the Platform Key" as described on page 18. These are your license keys.

   Very important: Do not copy the CMS address keys to multiple Sitecore instances, because this is a violation of the Clay Tablet License Agreement. Using the same CMS address keys on multiple Sitecore instances will cause the Connector to behave unexpectedly, which can result in lost translation content, orphaned projects, and inaccurate translation status reports. Clay Tablet will only support technical issues caused by duplicating or incorrectly installing CMS address keys on a time and materials basis.

6. If you are installing the Connector on Sitecore 6.x, you must set the .NET target framework in IIS for application pool to 4.0. This is necessary because the package has some newer assemblies that cannot target the earlier target framework.



Start Page
MUM-19375 (CORPNET\Kedar.Purandare)
Application Pools
Sites
Clean65
Default Web Site
EK861
Ektron85
Ektron86
Ektron90
GreetService
Sitecore81dev
SitecoreDev
Sitecoretraining
UCCAdministration

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker process

Filter: Group by: No Grouping

Name	Status	.NET Frame...	Managed Pipeli...	Identity	Applications
ASP.NET v4.0	Started	v4.0	Integrated	ApplicationPoolIden...	0
ASP.NET v4.0 Classic	Started	v4.0	Classic	ApplicationPoolIden...	0
ASP.NET v4.0 devEK90	Started	v4.0	Integrated	NetworkS	[illegible]
ASP.NET v4.0 EK861	Started	v4.0	Integrated	NetworkService	1
ASP.NET v4.0 EK91	Started	v4.0	Integrated	NetworkService	0
ASP.NET v4.0 EK9New	Started	v4.0	Integrated	NetworkService	0
ASP.NET v4.0 Ektron85	Started	v4.0	Integrated	NetworkService	1
ASP.NET v4.0 Ektron85sp2	Started	v4.0	Integrated	NetworkService	0
ASP.NET v4.0 Ektron86	Started	v4.0	Integrated	NetworkService	1
ASP.NET v4.0 Ektron90	Started	v4.0	Integrated	NetworkService	1
AuthorITApp	Started	v2.0	Integrated	NetworkService	3
Classic .NET AppPool	Started	v4.0	Classic	ApplicationPoolIden...	0
Clean65AppPool	Started	v4.0	Integrated	NetworkService	1

2.1 System Requirements

The Clay Tablet Connector for Sitecore (Connector) must be installed on the machine where Sitecore is installed. The Connector supports Sitecore versions 6.4 and higher, including all current releases.

The Connector has no additional hardware or software requirements beyond those of Sitecore CE. For detailed requirements, refer to the appropriate version of the *Sitecore CMS Installation Guide*. This guide is available in the documentation section of the Sitecore Developer Network (SDN) site, at http://sdn.sitecore.net/Reference.aspx. You must log in to this site to access the documentation.

The Connector supports the Mozilla Firefox and Google Chrome web browsers. The Connector supports Microsoft Internet Explorer 11 with Sitecore 7 and 8.

2.2 Setting Your System Date, Time, and Time Zone Correctly

The Clay Tablet Connector sends content to and receives content from the Clay Tablet Platform, which is hosted in the Amazon Web Services (AWS) environment. AWS requires any machines that connect to its applications to have the correct system time and date settings.

Important: Before proceeding, ensure that the system date, time, and time zone are set correctly on any systems that will run the Clay Tablet Connector. If these settings are incorrect, the following error message is displayed: `Error. The difference between the request time and the current time is too large.`

2.3 Downloading the Delivery Package

1. Download the Clay Tablet Connector ("Connector") delivery package from the link that Clay Tablet Technologies sends you. This is a `.rar` file, which is a zipped file that contains that contains the following five folders:

Folder Name	Description
Accounts	Two or more keys, which enable the Connector to access the storage and message-queue services on the Amazon S3 and SQS web services. Each key is in a separate XML file. - `source.xml`: The Connector uses this CMS address key to move the files your company exports from Sitecore to the Clay Tablet Platform. The Platform then forwards your exported files to your translation provider. If you have multiple Sitecore installations, you must have one key (file) for each environment. If you are using multiple translation providers, there is one key (file) for each provider. - `target.xml`: This is your company's platform key, for the Clay Tablet Platform. **Very important:** Do not copy the CMS address keys to multiple Sitecore instances, because this is a violation of the Clay Tablet License Agreement. Using the same CMS address keys on multiple Sitecore instances will cause the Connector to behave unexpectedly, which can result in lost translation content, orphaned projects, and inaccurate translation status reports. Clay Tablet will only support technical issues caused by duplicating or incorrectly installing CMS address keys on a time and materials basis.

Folder Name	Description
Documents	■ *Clay Tablet Connector for Sitecore Installation and Configuration Guide* (this document) ■ *Clay Tablet Connector for Sitecore User Guide* ■ `CT3_LanguageCodes.txt`: The Connector language codes. Later you will set up these codes for each Sitecore language your company uses for translation, both source and target languages. These language codes are also listed in "Appendix: Language Codes" on page 72.
Sitecore Package	`CT35_Sitecore6_Client-3.x.x.zip`, where *x.x* is the current version number of the Connector for Sitecore. This the installation package that you will install into your Sitecore Content Editor system.
Workbox	Advanced Workbox is an optional feature, which replaces the standard Sitecore Workbox. It enables much more powerful management of large lists of items, including by language. You can sort by different column headings, approve many items at once, and manage the display of long lists easily. **Important:** Install this feature only if you are comfortable with replacing the standard Sitecore workbox. This folder contains the following items: ■ Advanced Workbox installation package ■ Advanced Workbox Installation Guide

2. Unzip the delivery package file you downloaded, and save its contents to a convenient location.

2.4 Setting Up the Translation Database

You set up the database so that the Connector can use it to store translation-related data.

- For instruction on setting up a Microsoft SQL Server database, see page 15.
- For instructions on setting up an Oracle database, see page 16.
- For instructions on setting up a Microsoft Azure SQL database, see page 16.

2.4.1 Translation Database Size Requirements

The required size of the database depends on the volume of translation you anticipate. When the database is used only for translation-related data, 100MB is usually adequate. However, by default, the Clay Tablet Connector backs up translation data, which it uses to support the Translation Correction, Update Remote TM, and SyncTranslation features. For descriptions of these features, refer to the *Clay Tablet Connector for Sitecore User Guide*.

- If you keep the default setting of backing up data so that you can use these features, you may need to increase the size of the database by 50 to 100 percent to support the backup.
- If you do not want to back up data, you do not need to increase the size of the database.

Note: To prevent the Connector from backing up translation-related data, you modify the `ClayTablet.BackupTranslationData` setting, described in "Configuring Database Settings" on page 17.

You can use the Connector to delete the translation-backup database at any time from the Sitecore Content Editor, however the relevant Connector features will not work properly. For instructions on deleting the backup, refer to the *Clay Tablet Connector for Sitecore User Guide*.

2.4.2 Setting Up a Microsoft SQL Server-Based Translation Database

This section describes how to set up a Microsoft SQL Server database to use with the Connector. For information on database size requirements, see "Setting Up the Translation Database" on page 14.

1. Use Microsoft SQL Server Management Studio to create an empty database. For detailed instructions, refer to the Microsoft SQL Server documentation.

   Recommendation: Name the new database `CT3Translation`.

2. Configure the Connector database connection string. The `Website_root/App_Config/ConnectionStrings.config` file defines the database connections.

   Important: The database part of the connection string must match the database name you created in the previous step.

   Typically, it should be:

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <!--
    Sitecore connection strings.
    All database connections for Sitecore are configured here.
  -->
  <add name="core" connectionString="user id=xxx;password=xxxt;Data
  Source=xxx;Database=XXX_Core"/>
  <add name="master" connectionString="user id=xxx;password=xxxt;Data
  Source=xxxx;Database=XXX_Master"/>
  <add name="web" connectionString="user id=xx;password=xxx;Data
  Source=xxx;Database=XXX_Web"/>
</connectionStrings>
```

   You must add one more connection string called `CT3Translation` for the Connector database. This is the name of the database from above:

```
<add name="CT3Translation" connectionString="user id=xxx;password=xxx;Data
Source=xxx;Database= CT3Translation "/>
```

Note: Do not use any name other than `CT3Translation` for the connection string.

The `User ID`, `password`, and `Data Source` values are usually the same as those used for the other connection strings.

Note: The user specified in the connection string must have permission to modify the table structure in the database. This user must have permission to run `ALTER TABLE` statements.

2.4.3 Setting Up an Oracle-Based Translation Database

This section describes how to set up an Oracle database to use with the Connector. For information on database size requirements, see "Setting Up the Translation Database" on page 14.

Recommendation: Create a separate Oracle database user for Clay Tablet data. Alternatively, you can use an existing Oracle database user for Clay Tablet data.

1. Optional. Create a Oracle database user for Clay Tablet data, using SQL statements similar to the following:

   ```
   CREATE USER ctttrans IDENTIFIED BY xxxxxxxx
   DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;
   GRANT UNLIMITED TABLESPACE TO ctttrans;
   GRANT CONNECT, CREATE TABLE TO ctttrans;
   ```

2. Add a `CT3Translation.Oracle` connection to the Sitecore connection string configuration. Add the following to the Sitecore connection string configuration:

   ```
   <add name="CT3Translation.Oracle" connectionString="user
   id=ctttrans;password=xxxxxxxx;Data Source=XE"/>
   ```

   Depending on the active configuration in the user's setup, this is one of the following:

   - `Website/App_Config/ConnectionStrings.config`
   - `Website/App_Config/ConnectionStringsOracle.config`

   Note: The user specified in the connection string must have permission to modify the table structure in the database. This user must have permission to run `ALTER TABLE` statements.

3. Ensure that the right version of `Oracle.DataAccess.dll` is in the `Website/bin` directory or in the right path. The DLL should match the Oracle database that the user is using. If the user is already using Oracle as the database for the Sitecore content, this should already be set up correctly.

2.4.4 Setting Up a Microsoft Azure SQL-Based Translation Database

This section describes how to set up a Microsoft Azure SQL database to use with the Connector. For information on database size requirements, see "Setting Up the Translation Database" on page 14.

1. Create an empty Azure database. For detailed instructions, refer to the Microsoft Azure documentation.

   Recommendation: Name the new database `CT3Translation`.

2. Configure the Connector database connection string. The `Website_root/App_Config/ConnectionStrings.config` file defines the database connections.

 Important: The database part of the connection string must match the database name you created in the previous step.

 Typically, it should be:

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <!--
    Sitecore connection strings.
    All database connections for Sitecore are configured here.
  -->
  <add name="core" connectionString="user id=xxx;password=xxxt;Data
  Source=xxx;Database=XXX_Core"/>
  <add name="master" connectionString="user id=xxx;password=xxxt;Data
  Source=xxxx;Database=XXX_Master"/>
  <add name="web" connectionString="user id=xx;password=xxx;Data
  Source=xxx;Database=XXX_Web"/>
</connectionStrings>
```

 You must add one more connection string called `CT3Translation` for the Connector database. This is the name of the database from above:

```
  <add name="CT3Translation" connectionString="Data
  Source=tcp:SERVER.database.windows.net;Initial Catalog=DB_NAME;Integrated
  Security=False;User ID=USER_NAME@SERVER;Password=PASSWORD;Encrypt=True"/>
```

 For example:

```
  <add name="CT3Translation" connectionString="Data
  Source=tcp:mpr0fb9mzh.database.windows.net;Initial Catalog=CTT;Integrated
  Security=False;User ID=CTT@mpr0fb9mzh;Password=csct@blet99;Encrypt=True"/>
```

 Note: Do not use any name other than `CT3Translation` for the connection string.

 The `User ID`, `Password`, and `Data Source` values are usually the same as those used for the other connection strings.

 Note: The user specified in the connection string must have permission to modify the table structure in the database. This user must have permission to run `ALTER TABLE` statements.

2.4.5 Configuring Database Settings

You configure the Connector's database settings by modifying the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`. You can specify the following settings:

Setting Name	Description	Supported Values	Default Value
ClayTablet. Translation Database	The location of your Sitecore content. If your Sitecore content is not in the `master` database, then you must change this value to match the name of your content database.	a database name	master
ClayTablet. Backup TranslationData	Determines whether or not the Connector stores backup translation data on the Content Editor server to support the Translation Correction, Update Remote TM, and SyncTranslation features. For descriptions of these features, refer to the *Clay Tablet Connector for Sitecore User Guide*. If you do not want to use these features, you can change this setting to `False`. This prevents substantial amounts of data from being stored on your Sitecore server, but it also prevents your company from accessing these useful features. **Recommendation:** Initially keep the default setting of `True` in case you want to use any of the features that rely on this data.	- True - False	True

2.5 Creating the Connector Folders

This section describes how to create folders that the Connector uses to store licensing information and translation data.

1. Under the Sitecore data folder, create the `CT3` folder.

   Important: Ensure that you create the `CT3` folder under the Sitecore data folder, and not under the Sitecore website root folder. To locate the Sitecore data folder, check the `Web.config` file, searching for a line such as:
   `<sc.variable name="dataFolder" value="C:\Inetpub\wwwroot\Sitecore6\Data\"/>`.
   For the location in this example, you would create the `CT3` folder as:
   `"C:\Inetpub\wwwroot\Sitecore6\Data\CT3"`.

2. Create two sub-folders under the `CT3` folder you just created:
   - `Accounts`. Set folder security so that the Windows account used by IIS has read permission.
   - `Data`. Set folder security so that the Windows account used by IIS has full permission.

2.6 Installing the CMS Address Key and the Platform Key

Very important: Do not copy the CMS address keys to multiple Sitecore instances, because this is a violation of the Clay Tablet License Agreement. Using the same CMS address keys on multiple Sitecore instances will cause the Connector to behave unexpectedly, which can result in lost translation content, orphaned projects, and inaccurate translation status reports. Clay Tablet will only support technical issues caused by duplicating or incorrectly installing CMS address keys on a time and materials basis.

- The Connector uses the CMS address key, which is in `source.xml`, to move the files your company exports from Sitecore to the Clay Tablet Platform. The Platform then forwards your exported files to your translation provider. If you have multiple Sitecore installations, there must be one key (file) for each environment. If you are using multiple translation providers, there must be one key (file) for each provider.

  Note: The Connector always initiates calls to the Clay Tablet Platform. However, the CMS address keys enable establishing a secure, discrete connection between the Connector instance and the Platform. They also support the return of files from translation.

- Your company's platform key is in `target.xml`. This is your company's license for the Clay Tablet Platform.

  Important: If you do not install this key, the Connector will run. However, an **Upgrade** button will be displayed in the **CTC Translation** tab in the Sitecore Content Editor ribbon, and many user interfaces will display errors.

To install the CMS address key and the platform key:

- Copy the file with the CMS address key (`source.xml`) and the file with the platform key (`target.xml`) from the `<Delivery Package/Accounts>` folder to the `CT3/Accounts` folder. This installs the license for one Sitecore instance and one translation provider.

  Note: For instructions on installing CMS address keys for additional service providers, see "Configuring the Connector to Work with Multiple Translation Providers" on page 23.

Now you are ready to install the Connector, as described in "Installing the Clay Tablet Connector for Sitecore" on page 20.

3 Installing the Clay Tablet Connector for Sitecore

This section describes how to install the Connector installation package you downloaded earlier into your Sitecore system.

Before you install the Clay Tablet Connector ("Connector"), verify that you have reviewed the system requirements and followed all the pre-installation procedures described in "Before You Install" on page 12.

To install the Connector into Sitecore:

1. On the Sitecore Desktop, on the Windows Start menu, select **Sitecore > Development Tools > Installation Wizard**.

   The **Welcome** page of the **Sitecore Install Package** wizard opens.

2. Click **Next**.

   The **Select Package** page of the **Sitecore Install Package** wizard opens.

3. Click **Browse**, and locate the installation package, `CT35_Sitecore6_Client-3.x.x.zip`, where *x.x* is the current version number of the Connector for Sitecore that you downloaded and unzipped earlier.

   Tip: The installation package is in the following location in the delivery package: `<Delivery Package/Sitecore Package/CT35_Sitecore6_Client-3.x.x.zip>`.

4. Click **Upload** to upload the package.

5. After the package is uploaded, click **Next**.

6. Follow any on-screen instructions to finish installing the Connector package into Sitecore.

Next you configure Connector parameters, as described in "Configuring the Clay Tablet Connector for Sitecore" on page 21.

4 Configuring the Clay Tablet Connector for Sitecore

You perform the following steps to configure your Connector installation:

1. "Configuring the Connector for the Clay Tablet On-Premise Platform" on page 21. Required only if you are using the Clay Tablet On-Premise Platform ("Platform"), which is the Clay Tablet Platform that you host and manage, in contrast to the Clay Tablet Platform that is hosted on AWS by Clay Tablet.
2. "Configuring the Connector to Run in a Clustered Environment of Multiple Load-Balanced Sitecore Servers" on page 22
3. "Configuring Network Settings for a Firewall" on page 22. Optional.
4. "Configuring the Connector to Work with Multiple Translation Providers" on page 23. Required only if you will be sending content for translation to multiple translation providers.
5. "Configuring Global Translation Settings" on page 24.
6. "Configuring Global Service Settings" on page 39. Optional.
7. "Configuring the Sitecore Languages in the Sitecore Content Editor" on page 41.
8. "Configuring the Translation Workflow" on page 42.
9. "Filtering Fields in Items that Do Not Need Translation" on page 51.
10. "Configuring Connector Roles and Adding Users" on page 62.
11. "Configuring the In-Context Preview Feature" on page 63.

    Important: This feature is currently in alpha testing. It is not currently generally available for production usage.

4.1 Configuring the Connector for the Clay Tablet On-Premise Platform

If you are using the Clay Tablet On-Premise Platform ("Platform"), which is the Clay Tablet Platform that you host on your premises, you configure your Connector to point to this platform.

To configure the Connector to point to your Clay Tablet On-Premise Platform:

1. Open the `Website_root/Website/App_Config/Include/CT3Translation.config` file for editing.
2. Uncomment the following setting:

```
<!-- Uncomment this to configure URL to CTWS backend for On-Premise platform
<setting name="ClayTablet.CTWS.URL" value="http://ctws.ctt-platform-int.com/"/>
-->
```

3. Modify the `ClayTablet.CTWS.URL` setting to the DNS, IP address, or host name of the where your CTWS service is running on IIS.

   Note: The address must include the protocol, such as `http://` or `https://`, for example, `http://clay-tablet.com`.

4. Save your changes.

4.2 Configuring the Connector to Run in a Clustered Environment of Multiple Load-Balanced Sitecore Servers

In a clustered environment of multiple load-balanced Sitecore servers, all servers can provide the Connector's user-interface features. However, only one server can send and receive translation jobs. This server is the *primary server*. You use the `ClayTablet.Sitecore.PrimaryServer` setting to specify which Sitecore server is the primary server.

- In a single-server environment with one Sitecore server, this setting is not relevant, so it is commented out by default.
- In a clustered environment, you must configure the Connector on each Sitecore server to point to the primary server. This ensures that only the primary server can send or receive translations.

To configure the Connector on each Sitecore server in a clustered environment:

1. Open the `Website_root/Website/App_Config/Include/CT3Translation.config` file for editing.
2. Locate the `ClayTablet.Sitecore.PrimaryServer` setting, and uncomment it.
3. Change the value of this setting to either the host name or the IP address of the primary server.

   Important: Enter the same value on each Sitecore server in your clustered environment.

4. Save your changes.

Important: Repeat these steps on each Sitecore server in your clustered environment.

Note: To change the primary server, update this value on the current primary server before updating it on all the other servers.

4.3 Configuring Network Settings for a Firewall

Optional step. If you have a firewall, you must configure your ports so that the Connector can communicate with the Clay Tablet Platform. The Connector must be able to communicate with the Clay Tablet Platform by initiating the following outbound network connections:

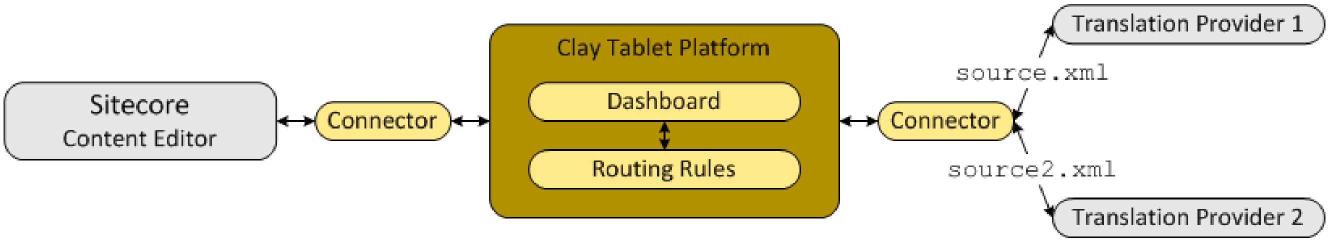
Protocol	Port Number	Description	Location
HTTP	Port 80	For access to Amazon's AWS S3 XML namespace and XSD file	http://s3.amazonaws.com
HTTP	Port 80	For access to Amazon's AWS SQS XML namespace and XSD file	http://queue.amazonaws.com
HTTPS	Port 443	For secure access to Amazon's AWS S3 service	https://s3.amazonaws.com
HTTPS	Port 443	For secure access to Amazon's AWS SQS service	https://queue.amazonaws.com

4.4 Configuring the Connector to Work with Multiple Translation Providers

This section is required only if your Connector will work with multiple translation providers. You can configure the Connector to work with multiple translation providers, so that you can send content to multiple translation vendors directly from Sitecore.

This section assumes that you have already installed the CMS address key (in `source.xml`) for your first (default) translation provider in step 4 of "Installing the Clay Tablet Connector for Sitecore" on page 20. The Connector requires a separate `source.xml` file for each translation provider, because the CMS address key is unique for each translation provider. The Connector uses this key to move your files from your Sitecore installation to the Clay Tablet Platform, and then to your translation provider.

Note: You use the same platform key (`target.xml`) for all your Sitecore installations and translation providers.



The instructions below describe how to install the CMS address key for your second translation provider:

1. Rename the file with the CMS address key for your second translation provider to `source2.xml`. This will prevent overwriting the CMS address key that you previously installed.

2. Copy the file with the CMS address key for your second translation provider to the following directory: `data/CT3/Accounts` on the Sitecore server.

3. Edit `Website/App_Config/Include/CT3Translation.config`. so that it is similar to the following:

   The default translation provider is configured with `ClayTablet.Account` (the name of the translation provider), `ClayTablet.SourceAccount` (CMS address key), and `ClayTablet.TargetAccount` (account license key):

```
<setting name="ClayTablet.Account" value="Translation Provider Name"/>
```

```
<setting name="ClayTablet.SourceAccount"
value="$(dataFolder/CT3/Accounts/source.xml"/>
```

```
<setting name="ClayTablet.TargetAccount"
value="$(dataFolder/CT3/Accounts/target.xml"/>
```

The second translation provider is configured with `ClayTablet.Account2` (the name of the translation provider) and `ClayTablet.SourceAccount2` (CMS address key):

```
<setting name="ClayTablet.Account2" value="Translation Provider 2 Name"/>
```

```
<setting name="ClayTablet.SourceAccount2" value="$(dataFolder)
/CT3/Accounts/source2.xml"/>
```

The value of `ClayTablet.TargetAccount2` (account license key) is generally the same for all translation providers.

Note: If this value is the same as ClayTablet.TargetAccount, you do not have to modify it.

```
<setting name="ClayTablet.TargetAccount2" value="$(dataFolder)
/CT3/Accounts/target.xml"/>
```

Repeat these steps for each additional translation provider that you work with.

Important: Ensure that the files with the CMS address keys have unique names so that they do not overwrite the CMS address keys for other translation providers.

4.5 Configuring Global Translation Settings

The Connector's global translation settings determine how the Connector sends out all translation jobs. These translation settings are in the following locations:

- the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`
- configuration settings in the Content Editor in: `/sitecore/System/Settings/Clay Tablet Settings`

You can perform the following configuration steps:

- "Configuring Bulk Translation Settings" on page 25
- "Configuring Email-Notification Settings" on page 29
- "Configuring Logging Settings" on page 33
- "Configuring Platform Settings" on page 33
- "Configuring Target Translation Data Settings" on page 34
- "Configuring Translation Settings" on page 34
- "Configuring Update-TM Settings" on page 38

4.5.1 Configuring Bulk Translation Settings

There are multiple places to configure settings for the Bulk Translation wizard:

Setting	Where to Configure	For Details, See...
■ the default value of the **Sync Translation** check box in the **Advanced Translation Options** section of the **Translation Options** page in the wizard ■ which target languages to display in the **Settings** page ■ the default value of the **Send Dependent Items** check box in the **Root of Source Item** page of the wizard ■ whether to collect Sitecore items in specific templates, including item buckets, to send out for translation	the Sitecore user interface	"Configuring Bulk Translation Settings in Sitecore" on page 27
all other settings	the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file	"Configuring Bulk Translation Settings in the Configuration File" on page 25

Note: You can also create an extension to modify the behavior of the **Send Dependent Items** check in the **Root of Source Item** page of the wizard. For details, see "Configuring the Behavior of the Send Dependent Items check box in the Bulk Translation Wizard" on page 81.

4.5.1.1 Configuring Bulk Translation Settings in the Configuration File

You can specify the following settings in the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file:

Setting Name	Description	Supported Values	Default Value
ClayTablet.Bulk Translation RootID	This sets the `/Sitecore/content` directory (ID: {0DE95AE4-41AB-4D01-9EB0-67441B7C2450}) as the default root from which the Connector collects items for translation for the Bulk Translation feature. This feature collects large batches of content from the site tree for bulk export. If you want content from only one site to be translated, then you must change the value of this item's ID to that path. For example if you want to translate items only from `Sitecore/Content/home/site1/`, then you must change the value of this item's ID to `Sitecore/Content/home/site1/`. **Note:** There can be only one root directory.	either the default ID or a path	{0DE95AE4-41AB-4D01-9EB0-67441B7C2450}
ClayTablet. Maximum ItemsInOne TranslationFile	This setting determines the maximum number of Sitecore items to pack into a single file for translation. You can adjust the number to better suit your translation requirements. **Recommendation:** Discuss this with your translation provider.	an integer	100
ClayTablet. PreventSenting ItemsWithout Workflow	When using the Bulk Translation wizard, determines whether or not the Connector sends out items for translation that are not assigned to a workflow. ■ If you set this value to `True`, then when using the Bulk Translation wizard to send out content for translation, the Connector does not send out items not assigned to a workflow. ■ If you set this value to `False`, then using the Bulk Translation wizard sends out all selected items for translation, even if they are not assigned to a workflow. **Warning:** This creates a target version with source language content, which becomes publishable immediately.	■ True ■ False	■ `True` for new Connector installations ■ `False` for upgraded Connector installations

Setting Name	Description	Supported Values	Default Value
ClayTablet.StateToAssignInBulkTranslation	Determine the default state displayed in **Workflow Option: applied to all items selected > With state** dropdown list in the Translation Options page of the Bulk Translation Wizard. This is the default state to assign to source items without workflow state when sending them out for translation from the wizard. The default value points to the Review state in `CT3SampleWorkflow`, but you can change this to another state in that workflow or another workflow.	a workflow state	4B7E2DA9-DE43-4C83-88C3-02F042031D04

4.5.1.2 Configuring Bulk Translation Settings in Sitecore

You can specify some settings for the Bulk Translation wizard in the Sitecore user interface.

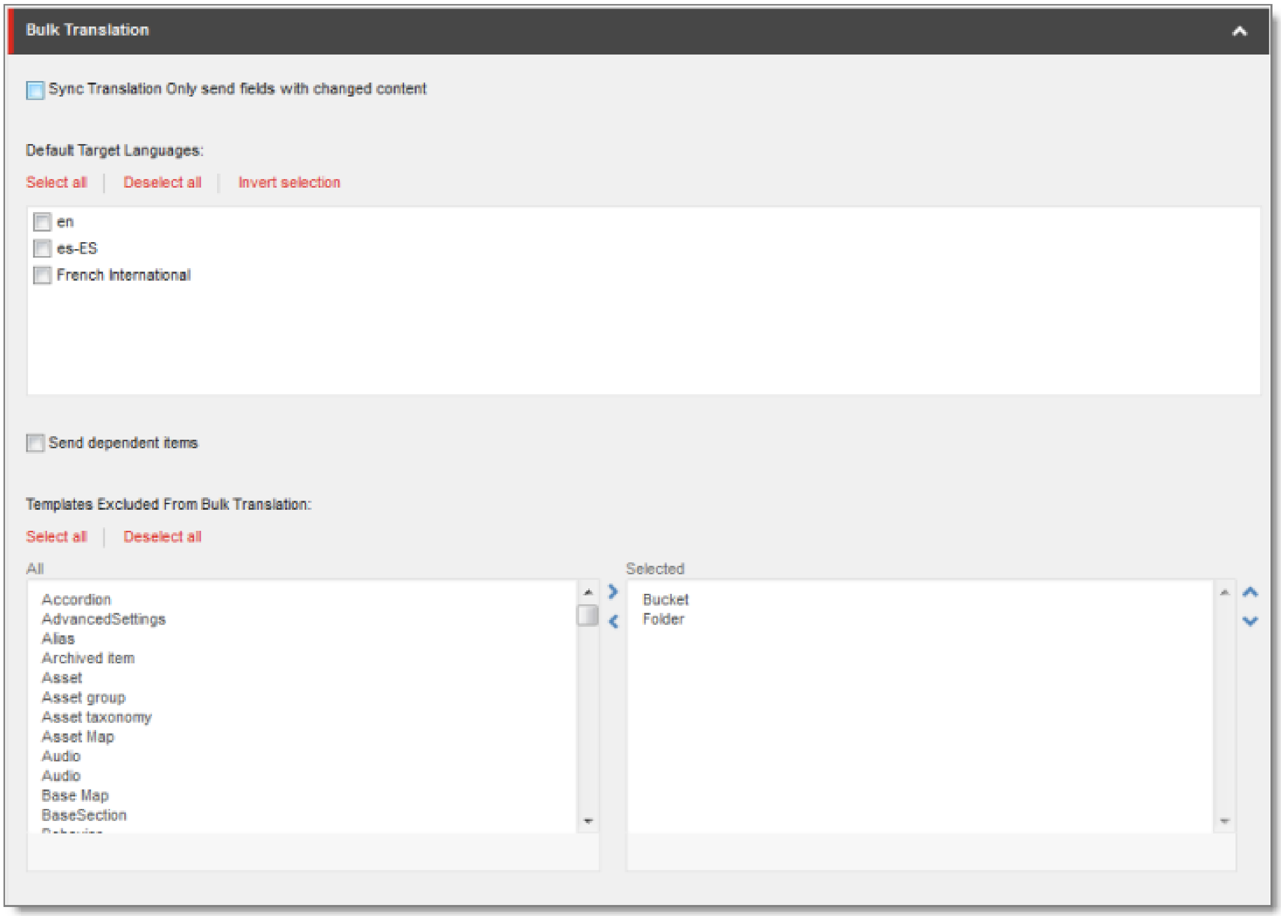
To configure settings for the Bulk Translation wizard:

1. In the Content Editor, in the content tree, navigate to the `/sitecore/system/Settings/Clay Tablet Settings` folder.

   Tip: Scroll down to the bottom of the `Settings` folder to locate the `Clay Tablet Settings` sub-folder.

2. Click the **Clay Tablet Settings** item in the folder to open it in the Content Editor.

3. Scroll down to the **Bulk Translation** section.



Bulk Translation

☐ Sync Translation Only send fields with changed content

Default Target Languages:

Select all | Deselect all | Invert selection

☐ en
☐ es-ES
☐ French International

☐ Send dependent items

Templates Excluded From Bulk Translation:

Select all | Deselect all

All

Accordion
AdvancedSettings
Alias
Archived item
Asset
Asset group
Asset taxonomy
Asset Map
Audio
Audio
Base Map
BaseSection

Selected

Bucket
Folder

4. Specify one or more of the following options:

Option	Description
Sync Translation Only send fields with changed content	Select or clear this check box to specify the default value of the **Sync Translation** check box in the **Advanced Translation Options** section of the **Translation Options** page in the wizard. For a detailed explanation of this option, refer to "Sending Multiple Content Items for Translation" in the *Clay Tablet Connector for Sitecore User Guide*.
Default Target Languages	In this section, you can select which target languages to select by default in the **Settings** page of the wizard by selecting the corresponding check boxes. If you do not select any target languages, then all the check boxes for the target languages are blank.
Send dependent items	Select or clear this check box to specify the default value of the **Send Dependent Items** check box in the **Root of Source Item** page of the wizard.

Option	Description
Templates Excluded from Bulk Translation	Use the arrows to move items to exclude to the **Selected** list. This instructs the wizard to exclude Sitecore items with the corresponding templates when it collects items for translation. By default, **Bucket** and **Folder** are selected, so that the wizard excludes item buckets and folders from items that it collects to send out for translation.

4.5.2 Configuring Email-Notification Settings

There are multiple places to configure email-notification settings:

Setting	Where to Configure	For Details, See...
The default amount of time, in hours, that a job is stuck, and it fails to reach `10%` (`In Translation`) status. After this interval, the Connector automatically sends an email notification that the job is stuck.	the Sitecore user interface	"Configuring Email Notifications in Sitecore" on page 32
all other settings	the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file	"Configuring Email Notifications in the Configuration File" on page 29

4.5.2.1 Configuring Email Notifications in the Configuration File

You can specify the following settings in the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file:

Setting Name	Description	Supported Values	Default Value
ClayTablet.EmailNotification.SendEmail	Determines whether or not the Connector sends email notifications when it sends items out for translation or it receives translated items back from translation. - To receive email notifications, keep this setting as `True`. - To prevent receiving email notifications, change this setting to `False`.	- True - False	True

Setting Name	Description	Supported Values	Default Value
ClayTablet. EmailNotification. WaitMinutes	The Connector does not send a separate email notification for each item sent for translation. This setting determines the interval, in minutes, that the Connector waits to send one notification email. That email contains information about all the items that were sent for translation since the previous notification.	integer	4
ClayTablet. EmailNotification. SmtpService	Determines which SMTP service the Connector uses to send notification emails: - `LocalSmtp`: The Connector uses the SMTP service on a local server to send emails. - `Gmail`: If your local server does not have an SMTP service, the Connector uses the free Gmail SMTP service to send emails.	- LocalSmtp - Gmail	LocalSmtp
ClayTablet. LOCAL.SMTP. Server	The IP address of the local SMTP service that sends notification emails. Relevant only if the value of the `ClayTablet.EmailNotification.SmtpService` setting, described above, is `LocalSmtp`.	an IP address	127.0.0.1
ClayTablet. LOCAL.SMTP. Port	The port number of the local SMTP service that sends notification emails. Relevant only if the value of the `ClayTablet.EmailNotification.SmtpService` setting, described above, is `LocalSmtp`.	a port number	25
ClayTablet. LOCAL.SMTP. EnableSSL	Determines whether the SMTP server uses SSL.	- True - False	False
ClayTablet. LOCAL.SMTP. FromAddress	The "From" email address that the local SMTP service uses to send notification emails. Relevant only if the value of the `ClayTablet.EmailNotification.SmtpService` setting, described above, is `LocalSmtp`.	an email address	CttNotification @Clay-Tablet.com

Setting Name	Description	Supported Values	Default Value
ClayTablet. LOCAL.SMTP. User	The username for authentication for the SMTP server, if required. **Note:** To use SMTP authentication, uncomment this section.	username	blank
ClayTablet. LOCAL.SMTP. Password	The password for authentication for the SMTP server, if required. **Note:** To use SMTP authentication, uncomment this section.	password	blank
ClayTablet. EmailNotification. NotifySentOut	Determines whether or not the Connector sends email notifications when it sends items out for translation. - To receive email notifications, keep this setting as `True`. **Note:** You must also set the `ClayTablet.EmailNotification.SendEmail` setting, described above, to `True`. - To prevent receiving email notifications, change this setting to `False`.	- True - False	True
ClayTablet. EmailNotification. NotifySentOut. Users.RoleName	The Connector sends email notifications when it sends items out for translation to all users that have been added to this role. This role is installed by default. You must add users to this role, or change the role name and this setting to fit your requirements. For information about roles, see "Configuring Connector Roles and Adding Users" on page 62.	a role name	sitecore\ Translation ItemSent Notification Receiver
ClayTablet. EmailNotification. NotifyCompleted	Determines whether or not the Connector sends email notifications when it receives translated items back from translation. - To receive email notifications, keep this setting as `True`. **Note:** You must also set the `ClayTablet.EmailNotification.SendEmail` setting, described above, to `True`. - To prevent receiving email notifications, change this setting to `False`.	- True - False	True

Setting Name	Description	Supported Values	Default Value
ClayTablet. EmailNotification. NotifyCompleted. Users.RoleName	The Connector sends email notifications when it receives translated items back from translation to all users that have been added to this role. This role is installed by default. You must add users to this role, or change the role name and this setting to fit your requirements. For information about roles, see "Configuring Connector Roles and Adding Users" on page 62.	a role name	sitecore\ Translation ItemCompleted Notification Receiver
ClayTablet. EmailNotification. NotifyErrors	Determines whether or not the Connector sends notification emails when an error occurs related to a translation job.	■ True ■ False	True
ClayTablet. EmailNotification. NotifyErrors. Users.RoleName	Sitecore users assigned to this role will receive notification emails. You can change the role name to fit your business, or you can add users to this default role.		Translation Administrator (installed by the Connector)

4.5.2.2 Configuring Email Notifications in Sitecore

You can specify an email-notification setting in the Sitecore user interface.

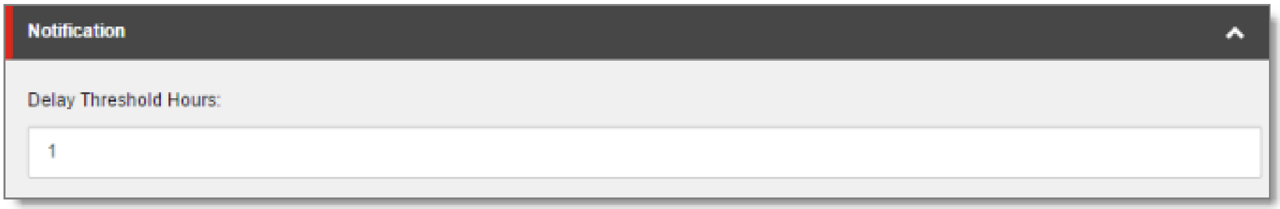
To configure an email-notification setting:

1. In the Content Editor, in the content tree, navigate to the `/sitecore/system/Settings/Clay Tablet Settings` folder.

   Tip: Scroll down to the bottom of the `Settings` folder to locate the `Clay Tablet Settings` sub-folder.

2. Click the **Clay Tablet Settings** item in the folder to open it in the Content Editor.

3. Scroll down to the **Notification** section.



Delay Threshold Hours:

1

4. Specify the following option:

Option	Description
Delay Threshold Hours	Enter the number of hours after which sending out a translation job is considered "stuck." If a job is still not at 10% (In Translation) status after this interval, the Connector sends out an email notification.

4.5.3 Configuring Logging Settings

You configure the Connector's logging settings by modifying the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`. You can specify the following settings:

Setting Name	Description	Supported Values	Default Value
ClayTablet.LogFolder	By default, the Connector logs are in the same folder as the Sitecore logs. This enables you to use the Sitecore Log Viewer tool to view Connector logs as well. Connector log files use the following file-name format: `log.CTTLOG.yyyymmdd.txt`.	a path	$(dataFolder/Logs/
ClayTablet.LogLevel	The minimum logging level written to the log files.	From the highest to lowest log level: ■ ERROR ■ INFO ■ DEBUG ■ Diagnose	INFO

4.5.4 Configuring Platform Settings

You configure the Connector's setting for the Clay Tablet Platform by modifying the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`. You can specify the following setting:

Setting Name	Description	Supported Values	Default Value
ClayTablet.HeartBeatIntervalInMinutes	Determines how frequently, in minutes, the Connector sends a heartbeat message to the Clay Tablet Platform that enables the Platform to monitor the health of the Connector. To disable the heartbeat, set this to zero (0).	integer	30

4.5.5 Configuring Target Translation Data Settings

You configure the Connector's settings for target translation data by modifying the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`. You can specify the following settings:

Setting Name	Description	Supported Values	Default Value
ClayTablet. OnlyKeep TranslationData FromLatest Version	Determines whether the Connector keeps translation data only for the latest version of a content item. - To keep translation data only for the latest version of a content item, keep this setting as `True`. - To keep translation data for all versions of a content item, change this setting to `False`.	- True - False	True
ClayTablet. TargetTranslation KeepAllData	When the Connector sends out content for translation, this setting determines which field values it copies to the target location. - To copy the field values from the previous version (if it exists) of the target items to the target location, change this setting to `True`. - To copy the field values from source items to the target location, keep this setting as `False`. **Note:** You specify the setting for copying the values of non-translatable fields separately, in the `TargetTranslationKeepNotTranslatableData` setting, as described below.	- True - False	False
ClayTablet. TargetTranslation KeepNot TranslatableData	When the setting of `TargetTranslationKeepAllData`, described above is `False`, then when the Connector sends out content for translation, this setting determines which values of non-translatable fields it copies to the target location. - To copy the values of non-translatable fields from the previous version (if it exists) of the target items to the target location, change this setting to `True`. - To copy the values of non-translatable fields from source items to the target location, keep this setting as `False`.	- True - False	False

4.5.6 Configuring Translation Settings

There are multiple places to configure translation settings:

Setting	Where to Configure	For Details, See...
which standard fields to copy to target items	the Sitecore user interface	"Configuring Translation Settings in Sitecore" on page 37
all other settings	the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file	"Configuring Translation Settings in the Configuration File" on page 35

4.5.6.1 Configuring Translation Settings in the Configuration File

You configure the Connector's translation settings by modifying the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`. You can specify the following settings:

Setting Name	Description	Supported Values	Default Value
ClayTablet.PackedFileFormat	The Connector can export Sitecore content (field content) in two different formats: XML and HTML. XML is the default format, which is acceptable to most translation providers. If your translation provider has difficulties handling XML files and can handle only HTML files, then you can change this setting to `HTML`.	either `XML` or `HTML`	XML
ClayTablet.TranslateCommonFolderItem	Determines whether or not the Connector sends out for translation content items that are based on a common-folder template, located in: `Templates/Common/Folder`. This determines whether the Connector sends out the folder's fields, including the folder name, for translation.	■ True ■ False	False

Setting Name	Description	Supported Values	Default Value
ClayTablet. PreviewURL	Determines whether or not to include a preview URL in the XML file, which enables translators and reviewers to preview the content. You can include the following parameters in the preview: - `{id}`: Sitecore item ID of the translated item. - `{version}`: The source version of the translated item. - `{lang}`: The source language of the translated item. - `{targetversion}`: The target version created as result of the translation. - `{targetlang}`: The target language for the translation. - `{jobid}`: The Connector translation job ID. **Note:** You must replace `&` with `&amp`. Otherwise the configuration file will not be valid XML, for example: `<setting name="ClayTablet.PreviewURL" value="http://www.yourcompany.com/preview?id={id}&amp;language={lang}&amp;version={version}&amp;jobid={jobid}" />` **Note:** To configure this setting, uncomment this section.		
ClayTablet.Send JobMetadata	Determines the format for sending job metadata. - To send job metadata in a format compatible with the Connector for Sitecore version 3.6 or higher, set this value to `True`. - To send job metadata in a format compatible with the Connector for Sitecore version 3.5 or lower, set this value to `False`.	- True - False	- `True` for new Connector installations - `False` for upgraded Connector installations

Setting Name	Description	Supported Values	Default Value
ClayTablet. Remove Control Characters	Some translation providers and translation management systems may not be able to handle control characters, such as `0X10` and `0X13`. They may treat an XML file with these characters as invalid. This option instructs the Connector to automatically remove all control (non-printing) characters (`0x00-0x1F`) from an XML file before sending it out for translation. - To automatically remove all control (non-printing) characters from an XML file before sending it out for translation, change this setting to `True`. - To keep all control characters in an XML file sent out for translation, keep this default setting of `False`.	- True - False	False
ClayTablet. Translate DisplayName	Supports sending out an item's display name for translation.	- true - false	false
ClayTablet. Sitecore. PreserveTemporaryFiles	Supports saving the temporary, generated files in the following scenarios: - The Connector sends out files for translation (source folder). - The Connector sends out files for TM update. - The Connector receives translated files from the Platform (target folder). Usually the Connector deletes these temporary files when it finishes processing them. However, they are useful for debugging and general troubleshooting.	- true - false	false

4.5.6.2 Configuring Translation Settings in Sitecore

You can specify a translation setting in the Sitecore user interface.

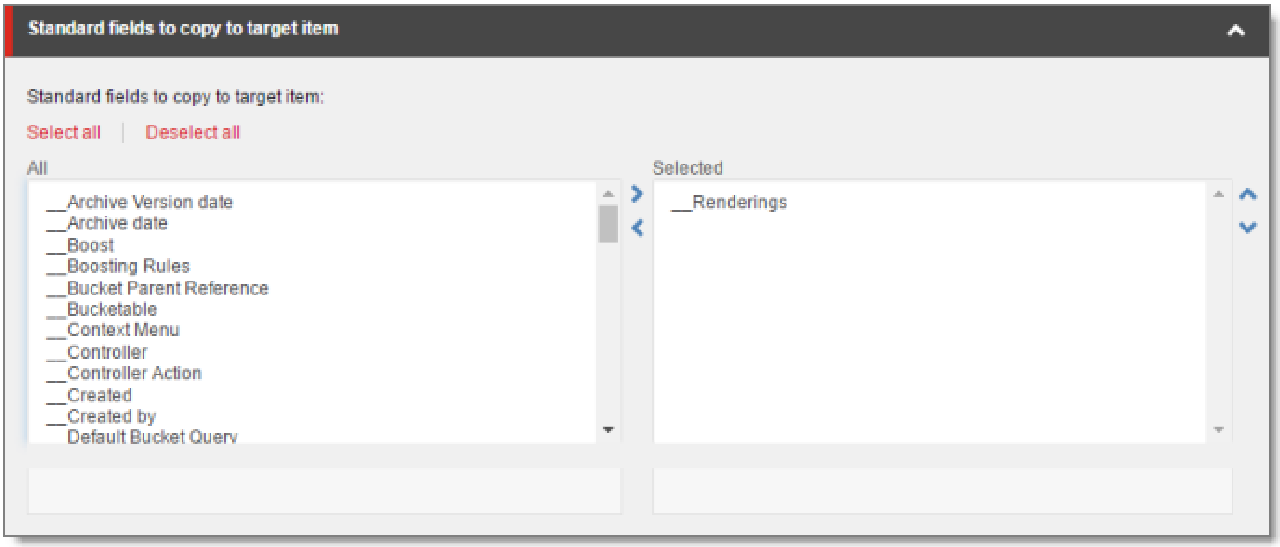
To configure a translation setting:

1. In the Content Editor, in the content tree, navigate to the `/sitecore/system/Settings/Clay Tablet Settings/Advanced Settings` folder.

   Tip: Scroll down to the bottom of the `Settings` folder to locate the `Clay Tablet Settings` sub-folder.

2. Click the **Advanced Settings** item in the folder to open it in the Content Editor.

3. Locate the **Standard fields to copy to target item** section.



Standard fields to copy to target item

Standard fields to copy to target item:

Select all | Deselect all

All

__Archive Version date
__Archive date
__Boost
__Boosting Rules
__Bucket Parent Reference
__Bucketable
__Context Menu
__Controller
__Controller Action
__Created
__Created by
__Default Bucket Query

Selected

__Renderings

4. If there are standard fields to copy to target items, select them in the **All** list and user the arrow to move them to the **Selected** list.

   Note: The standard fields are usually shared and therefore identical in all versions. Therefore, they are not usually copied or sent for translation.

4.5.7 Configuring Update-TM Settings

You can specify an Update-TM (translation memory) setting in the Sitecore user interface.

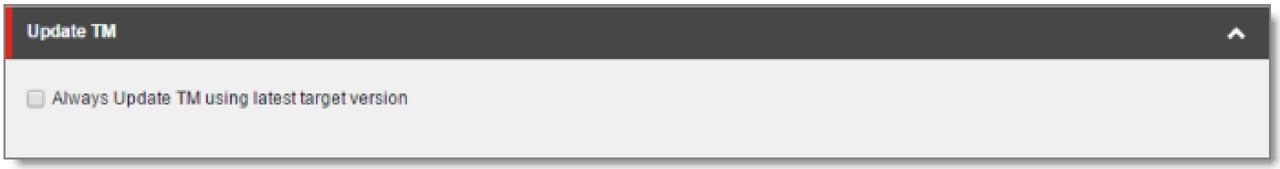
To configure Update-TM settings:

1. In the Content Editor, in the content tree, navigate to the `/sitecore/system/Settings/Clay Tablet Settings` folder.

   Tip: Scroll down to the bottom of the `Settings` folder to locate the `Clay Tablet Settings` sub-folder.

2. Click the **Clay Tablet Settings** item in the folder to open it in the Content Editor.

3. Scroll down to the **Update TM** section.



□ Always Update TM using latest target version

4. Specify the following option:

Option	Description
Always Update TM using latest target version	This determines which target version the Connector uses to send to update a remote TM (translation memory) server. ■ If this check box is cleared (default value), then the Connector always sends the target version that is the translation to the remote TM server. If the target version was edited post translation, then the Connector sends the version from the translation provider to the remote TM server. ■ If this check box is selected, then the Connector always sends the latest available target version to the remote TM server. If the target version was edited post translation, then the Connector sends this update of the translation to the remote TM server.

4.6 Configuring Global Service Settings

The Connector has two scheduled services, the Upload Service, and Download Service, which are defined in Sitecore scheduling. These services upload files to and download files from the Clay Tablet Translation Platform.

The default polling interval for these services is 2 minutes. You can change this interval by modifying the following configuration file: `Website_root/Website/App_Config/Include/CT3Translation.config`. The duration of the interval is expressed as `hh:mm:ss`.

Running these services has minimal impact on the performance of the Clay Tablet Translation Platform server, because if there are no active jobs when they contact the server, they just wait until the next scheduled invocation before contacting the server again. If the network is unavailable temporarily, these services log error messages on each invocation.

Setting Name	Description	Default Value
ClayTablet.CT3Agent.UploadService, ClayTablet.CT2Agent	This agent polls for translation jobs to send to the Clay Tablet Platform.	00:02:00
ClayTablet.CT3Agent.DownloadService, ClayTablet.CT2Agent	This agent polls for translation jobs to retrieve from the Clay Tablet Platform.	00:02:00

Note: To stop the Clay Tablet services from running, you can comment out the corresponding lines in the `CT3Translation.config` file.

Note: The Sitecore Scheduler controls all scheduled applications interacting with Sitecore, and therefore it schedules and triggers both these agents. The polling frequency of the agents cannot be more frequent than the Sitecore Scheduler's own frequency, because it will not have an impact.

Recommended polling intervals for these agents:

- Lowest = 1 minute
- Optimal = 5 minute

You can also configure the `UploadService` agent.

You can also configure `UploadService` to automatically send out all items in Translation Queue daily as single translation job.

Note: For instructions on configuring how `UploadService` creates translation jobs when automatically sending out items in the Translation Queue, see "Using Custom Logic to Configure how the UploadService Creates Jobs when Automatically Sending Out Items from the Translation Queue" on page 88.

To configure this setting so that the Connector automatically sends all items in the Translation Queue for translation daily:

1. Open `Website_root/Website/App_Config/Include/CT3Translation.config` for editing.
2. Locate the following line:

   ```
   <agent type="ClayTablet.CT3Agent.UploadService, ClayTablet.CT2Agent"
   method="Run" interval="00:02:00">
   ```

3. Uncomment the following lines, which follow:

   ```
   <AutoSendQueueItemsTime>hh:mm</AutoSendQueueItemsTime>
   <AutoSendQueueItemsJobName>AutoSendJob [{d} {t}]</AutoSendQueueItemsJobName>
   <AutoSendQueueItemsUser>sitecore\admin</AutoSendQueueItemsUser>
   <AutoSendQueueItemsLSP>LSP</AutoSendQueueItemsLSP>
   ```

4. Specify the following parameters in the uncommented lines:

Setting Name	Description
AutoSendQueueItemsTime	Specifies the time for sending the job from the Translation Queue out for translation, in the folloiwng format: `hh:mm`. You can use a delimiter to specify multiple times. The following example sends a new job at 6 AM and 6 PM daily: `<AutoSendQueueItemsTime>06:00;18:00 </AutoSendQueueItemsTime>`
AutoSendQueueItemsJobName	Configures the job name. - `{d}` is a placeholder for the day the job is sent. - `{t}` is a placeholder for the time the job is sent.
AutoSendQueueItemsUser	Optional. Specifies the username for sending job. If this is not configured, the user is `Anonymous`.
AutoSendQueueItemsUser	Optional. Specifies the translation provider where the Connector sends the job. If this is not configured, then the Connector sends the job to the default translation provider, as configured in the `ClayTablet.Account` setting in `Website/App_ Config/Include/CT3Translation.config`.

5. Save your changes.

4.7 Configuring the Sitecore Languages in the Sitecore Content Editor

The Connector uses the **Regional ISO code** field of the language item in Sitecore to define the Connector language codes. You must set up the correct Connector language codes for every Sitecore language your company uses for translation, both source and target languages. The Connector language codes are in the following location in the delivery package: `<Delivery Package/Documents/CT3_LanguageCodes.txt>`. These language codes are also listed in "Appendix: Language Codes" on page 72.

1. Log into the **Sitecore Content Editor** as an administrator.
2. Navigate to `sitecore/System/Languages`. This is where Sitecore languages are defined.

   Note: If a source or target language is not defined, you must add it to Sitecore. For detailed instructions, refer to the *Sitecore CMS Content Cookbook*. This guide is available in the documentation section of the Sitecore Developer Network (SDN) site, at http://sdn.sitecore.net/Reference.aspx. You must log in to this site to access the documentation.

3. For each source or target language, in the **Regional Iso Code field**, enter the Connector language code from the `<Delivery Package/Documents/CT3_LanguageCodes.txt>` file.

Note: The Connector includes error trapping for improperly configured language codes. If you make a mistake or forget to configure a language, when you try to send out content for translation, the language is not available for selection, and the `CT Language code is misconfigured` error is displayed beside the language.

4.7.1 Adding Custom Language Codes to Sitecore

You can also add custom language codes into Sitecore by editing the `Website/bin/CTAddedLanguageCodes.xml` file. However you must also contact Clay Tablet and your translation provider to ensure the custom languages are set up appropriately in the Clay Tablet Provider Connector as well as in the Translation Management systems.

Note: If your company sends content to your translation provider via the Clay Tablet FTP Connector on the translation side, it is not necessary to contact Clay Tablet, because the FTP Connector automatically creates folders based on the job name and language pairs, even for a custom language pair. However, you must inform your translation provider about the standard language code into which to translate the content.

4.8 Configuring the Translation Workflow

Sitecore uses workflows to control when a content item is published. The Connector uses workflows to control the translation process for a content item.

The Connector translation package installs a sample translation workflow, which you access in the Sitecore Content Editor as an administrator by navigating to `sitecore/System/Workflows/CT3 Sample Workflow`. Although you use this workflow to manage translation, you can duplicate, modify, or extend it to support additional requirements.

For general information about workflows, refer to the appropriate version of the *Sitecore CMS Content Author's Reference and Cookbook*. This guide is available in the documentation section of the Sitecore Developer Network (SDN) site, at http://sdn.sitecore.net/Reference.aspx. You must log in to this site to access the documentation.

The Connector supports translating Sitecore items that have been configured in the following ways:

- with the Connector Sample workflow attached
- with a custom workflow that includes the Connector configuration section and the associated workflow states, which are listed and described in "Connector Workflow States" on page 47
- with a custom workflow that does not have the Connector configuration section (see the following notes)
- without a workflow (see the following notes)

Note: You can use the Bulk Translation feature to submit all the above types of item for translation. However, if you want to send items for translation individually, the **Translate** button in the **Workflow** section of the **Review** tab is displayed only for Sitecore items attached to `CT3 Sample Workflow` or to another workflow whose base template is `TranslationWorkflow`, as described in "Using Your own Workflow for Translation" on page 48.

Note: You *must* use a translation workflow that includes the basic Connector translation steps, which are implemented through workflow states. This ensures that the Connector can send out your content and retrieve it from your translation provider(s). For information on Connector workflow states, see "Connector Workflow States" on page 47.

To configure the following settings, navigate to `sitecore/System/Workflows/CT3 SampleWorkflow` or your designated translation workflow.

Tip: Hover over each setting for extended help text.

You use the following settings to configure a workflow and the items attached to the workflow:

Note: For instructions on attaching items to a workflow, refer to the *Clay Tablet Connector User Guide*.

Parameter	Description	Default Value
Field Types to be Sent For Translation	Determines which types of fields the Connector can send out for translation when it sends an item for translation. This is relevant for all templates. If you create a custom field type, you must add it to this list so that you can send it out for translation. **Note:** You cannot set `Shared Fields` to be sent out for translation, because these fields are shared across all versions and languages of an item. To change these fields, click the **Edit** button above the list. In the dialog box that opens, select the field types to add, and use the blue arrow ➔ to copy them to the **Selected** column. When you are done, click **OK**. **Tip:** You can use the `Shift` key to select multiple adjacent field types.	By default, the Connector sends out fields of the following field types for translation: ■ text ■ Rich Text ■ html ■ Single-Line Text ■ Multi-Line Text
Translate From (Source Language)	The source language from which to translate the items. Select a language from the list.	Usually this is the Sitecore default language.
Translate To (Target Language)	The target languages that are available for translating the items attached to the workflow. To translate into specific target languages, select the corresponding check boxes.	n/a

Parameter	Description	Default Value
Synchronize Translation (detect changed content and resend)	Determines whether the SyncTranslation feature is available when translating items attached to this workflow. This feature checks each field of the source version for any content changes, and it sends *only* changed content fields out for translation, to ensure that all target versions reflect these changes. - When this check box is selected, you can activate this feature by selecting a check box in either the **Bulk Translation** wizard or the **Automatic Item Export for Translation** dialog box. The SyncTranslation feature may reduce translation costs, but it needs the Connector translation backup data, and it increases the Connector processing time before the Connector sends out the items for translation. - When this check box is cleared, you cannot activate this feature, because the related user interface is not displayed. The Connector sends all specified source content for translation, even if it has not changed since it was previously sent for translation.	check box is cleared
Requested Translation Timeframe (Days)	Determines the translation deadline when sending out the translation automatically, without using the user interface that supports entering this information in the **Delivery Date** field. This information is sent to the translation provider as metadata of the translation package. If you set this parameter to `14`, the deadline will be two calendar weeks after you submit an item for translation. **Recommendation:** Discuss whether to use this feature with your translation provider(s).	15
Workflow State to set when Sending Content for Translation	The workflow state to set when a source item has been selected for translation.	/sitecore/system/ Workflows/CT3 Sample Workflow/ Sending for Translation

Parameter	Description	Default Value
Workflow State to set (for Source Content) once content has been sent for translation	The workflow state to set for a *source* item after the Connector has sent it out for translation. For example, in some companies, when you send out the content for translation, the *source* is ready for publishing. In this scenario, you can set this state to `Done`, which is a publishable state. Alternatively, if your company has a final review process, you can set this to another state that is not publishable.	By default, this state is not set, so the source item's state does not change when you send it out for translation.
Workflow State to set for Content while it is out for translation	The workflow state to set for *target content* when an item is being translated. The default `In Translation` workflow state enables you to see which items are still out for translation. It facilitates following up with your translator about items that you have sent out for translation, but that you have not yet received back.	/sitecore/system/ Workflows/CT3 Sample Workflow/In Translation
Workflow State to set when a translation process has been cancelled	The workflow state to set when an item’s translation request has been cancelled. A user can cancel translation from the Translation Queue. A translation request is automatically cancelled if a fatal error occurs while sending out an item for translation, such as an incorrect language code, or insufficient user permissions.	/sitecore/system/ Workflows/CT3 Sample Workflow/ Reviewing
Workflow State to set when content Translation is complete	The workflow state to set when an item translation has been completed, and it is back in Sitecore for review and publishing.	/sitecore/system/ Workflows/CT3 Sample Workflow/ Editing Translated Version

Parameter	Description	Default Value
Send Content automatically using pre-configured language settings	Enables sending all items to the Translation Queue without further user action. That is, the Connector automatically uses the **Translate from**, **Translate to** and **Requested Translation Timeframe** information, defined above, to immediately send content to the Translation Queue without enabling the user to change these parameters: the Language Selection pop-up window does not open. - If this check box is selected, the **Language Selection** pop-up window does not open when you send content to the Translation Queue: you cannot select the target languages or the destination, but sending items for translation is streamlined. - If this check box is cleared, the **Language Selection** pop-up window opens when you send content out for translation: you can select the target languages, so the process is less streamlined. However, this provides greater control for the user.	check box is cleared
Automatically Publish content when it reaches the Final State	Determines whether the Connector automatically publishes the content when an item reaches the final state of the translation workflow. - If this check box is selected, the Connector automatically publishes the content when an item reaches the final state. - If this check box is cleared, the Connector does not automatically publish the content when an item reaches the final state.	check box is selected
Publish to which Database?	The target databases for publishing content if the Connector needs to publish an item.	n/a
Publish child items?	Determines whether the Connector publishes child items when it publishes their parent item. - If this check box is selected, the Connector publishes child items when it publishes their parent item. - If this check box is cleared, the Connector does not publish child items when it publishes their parent item.	check box is selected

For a list and description of Connector workflow states, see "Connector Workflow States", below.

4.8.1 Connector Workflow States

The `sitecore/System/Workflows/CT3 SampleWorkflow` workflow contains all required states for the Connector translation management process.

By default, it has the following seven states, which are listed, in order, below. Your company can modify this translation workflow to meet your business requirements. Your company can also create your own translation workflow, removing some of these states or adding additional states as required.

Workflow State	Relevant for Source Content	Relevant for Target Content
Editing	✓	
Reviewing	✓	
Sending for Translation	✓	
In Translation		✓
Editing Translated Version		✓
Reviewing Translated Version		✓
Done	✓	✓

Note: Only content in the `Done` state is publishable.

Source content

A source content item can move through the workflow in several different ways, depending on the configuration, which depends on your company's translation processes:

Example 1

Editing > Reviewing > Sending for Translation > Done

Example 2

Editing > Reviewing > Sending for Translation > Reviewing > Done

Target content

Similarly, a target content item can move through the workflow in several different ways, depending on the configuration, which depends on your company's translation processes:

Example

In Translation > Editing Translated Version > Reviewing Translated Version > Done

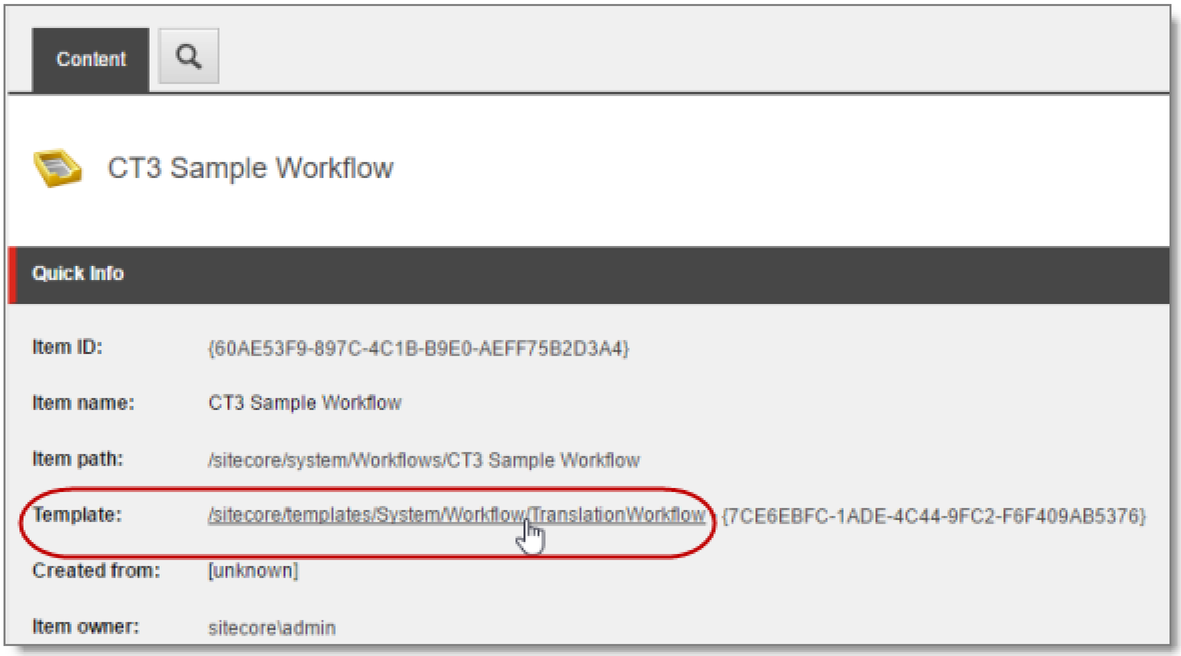
Note: When the state of a target content item is `In Translation`, that content is a copy of the source language item. Users can use the `In Translation` and `Editing Translated Version` workflow states to keep track of which translations have returned and which are still pending.

4.8.2 Using Your own Workflow for Translation

The sample translation workflow, `sitecore/System/Workflows/CT3 Sample Workflow` contains all the fields that the Connector requires. It contains both translation settings and some information about states and how the workflow proceeds between states. For a detailed description of these fields, see "Connector Workflow States" on page 47.

There is a hierarchy of templates in Sitecore:

The sample translation workflow is actually a template for the workflow provided in the Connector. It links to its parent template in the **Quick info** section:



Item ID: {60AE53F9-897C-4C1B-B9E0-AEFF75B2D3A4}

Item name: CT3 Sample Workflow

Item path: /sitecore/system/Workflows/CT3 Sample Workflow

Template: /sitecore/templates/System/Workflow/TranslationWorkflow {7CE6EBFC-1ADE-4C44-9FC2-F6F409AB5376}

Created from: [unknown]

Item owner: sitecore\admin

The parent of this is the `/sitecore/templates/System/Workflow/TranslationWorkflow` base template. If you click the link and navigate to that template, you can see its base templates:

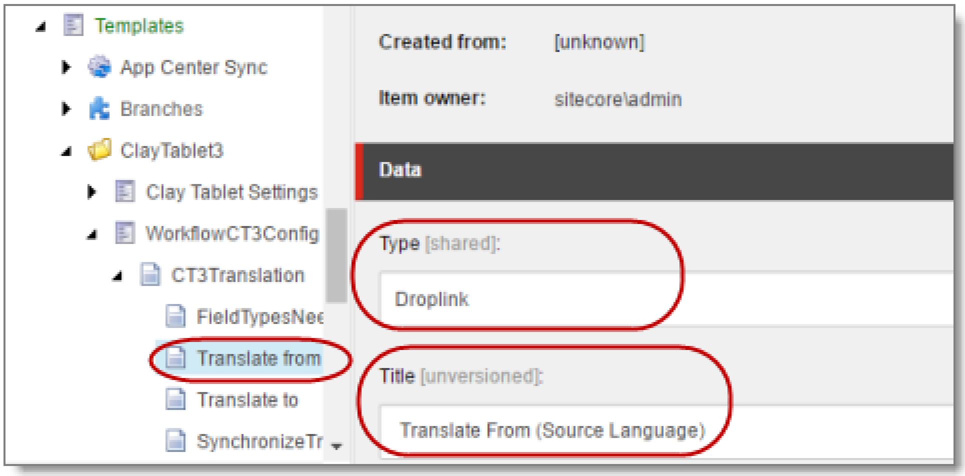
- `Standard template` -- This is a standard Sitecore template.
- `WorkflowCT3Config` -- This is the Connector's top-level workflow template. It is at the top of the hierarchy.

4.8.3 Modifying Controls in a Workflow

The Templates/ClayTablet3/WorkflowCT3Config template defines the controls in the CT3 Sample Workflow workflow.

For example:

- The **Translate from** leaf in the tree represents the control for defining source languages for translation.
- The **Type** field represents the type of user-interface control used to specify the source languages.
- The **Source** field defines the location where the Connector retrieves source languages in Sitecore.



Created from: [unknown]

Item owner: sitecore\admin

Data

Type [shared]:

Droplink

Title [unversioned]:

Translate From (Source Language)

There are multiple ways to modify controls in a workflow:

- You can modify the Templates/ClayTablet3/WorkflowCT3Config template.
- You can copy the values from the Templates/ClayTablet3/WorkflowCT3Config template into your own template.

There are many ways to implement modifications:

- You can remove a control. For example, if you decide you do not need a particular field, you can remove it from WorkflowCT3Config, using the **Builder** tab. (The **Content** tab is the default tab.)
- You can modify a control. For example, if you want to change a control, make a copy of the Templates/ClayTablet3/WorkflowCT3Config template, and then make the modifications you want.
- You can modify your own workflow by selecting **Insert > Insert from template** in the context menu, selecting the Templates/ClayTablet3/WorkflowCT3Config template, and then selecting a control.

4.8.4 Modifying States in a Workflow

You can modify the states in a workflow and the linkages between states, which determines the workflow.

Warning: Do not modify the values in the **Data** section, **Type string** field of states in `CT3 Sample Workflow` or in states that originate from there. Otherwise, the Connector will not work property. The values in that field are Clay Tablet classes that call the Connector functionality.

You can add or remove states from the template, or modify how the linkages between states, such as which state follows another state. There are two recommended ways to do this:

- **Modify states in CT3 Sample Workflow:** In the `CT3 Sample Workflow` folder, right-click and select **Insert > Insert from template** in the context menu. Navigate to the `Workflow/State` template. All the states are based on this template. Then, after you create the state within the workflow, you create an action. For example:
  - To specify the *next* state, in the **Data** section, select the next state in the **Next state** dropdown list.
  - To specify the *final* state, in the **Data** section, select the **Final** check box. By default, the final state is the `Publish` state.

  Tip: First create all the states and then link them, as described above.
- **Copy a state to a new workflow:** You can create a new state based on an existing state. To do this, select the state you want to want to copy. Right-click and select **Copying > Copy To** in the context menu. Then select the workflow where you want to copy this state, and click **Copy**. The copied state has the same settings as the original state.

For more information, refer to the *Sitecore CMS Workflow Cookbook*. This guide is available in the documentation section of the Sitecore Developer Network (SDN) site, at http://sdn.sitecore.net/Reference.aspx. You must log in to this site to access the documentation.

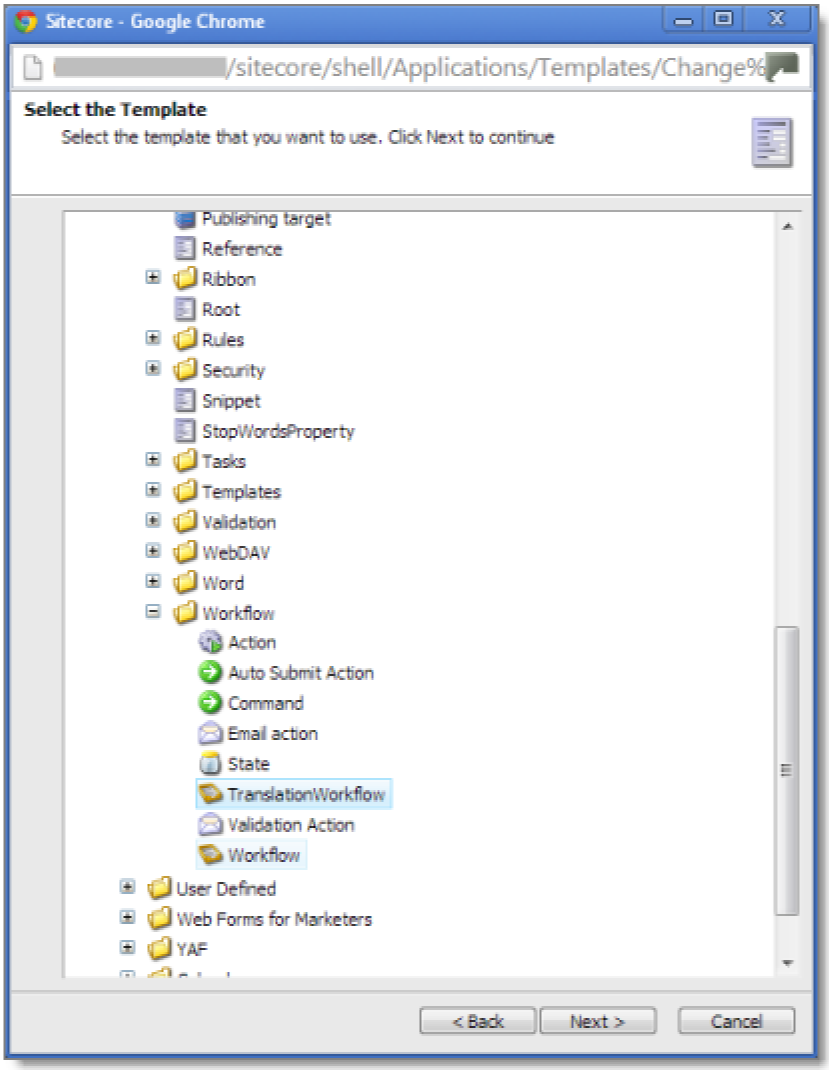
Note: Your custom workflow must include the required translation statuses.

4.8.5 Changing the Base Template of Your Workflow

To change the base template of your workflow, so that you can use it with the Connector to manage translations:

1. In the navigation pane, select your workflow that you want to use as a translation workflow. For example, select `sitecore\System\Workflow\MyWorkflow`.
2. In the ribbon, click the **Configure** tab.
3. In the **Template** section of the Configure tab, click **Change**.

   The Change Template Wizard Welcome page opens.
4. Click **Next**.
5. In the **Select the Template** page of the wizard, navigate to `Templates\System\Workflow\TranslationWorkflow`, and click **Next**.



Select the Template

Select the template that you want to use. Click Next to continue

6. In the Change page of the wizard, click **Next** to confirm this change.

7. Click **Finish**.

4.9 Filtering Fields in Items that Do Not Need Translation

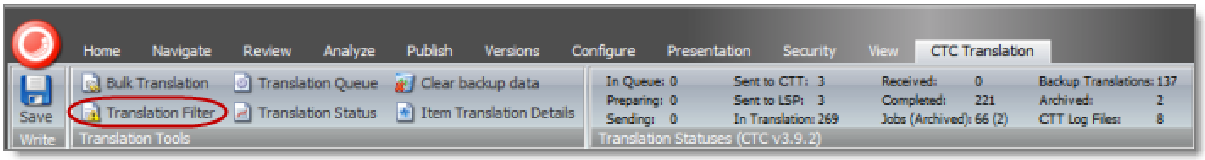
You can use the **Translation Filter** window to filter out fields that never need translation. You can access this feature from either the **Sitecore Content Editor** or the **Sitecore Desktop**.

You initially specify which types of fields the Connector sends for translation using the **Field Types to be Sent For Translation** workflow-level parameter. For details, see "Configuring the Translation Workflow" on page 42.

You use the Translation Filter feature to further exclude certain types of fields from content that the Connector sends for translation, per the template. You can create filters when initially configuring the Connector, or you can create or update the filters at any time.

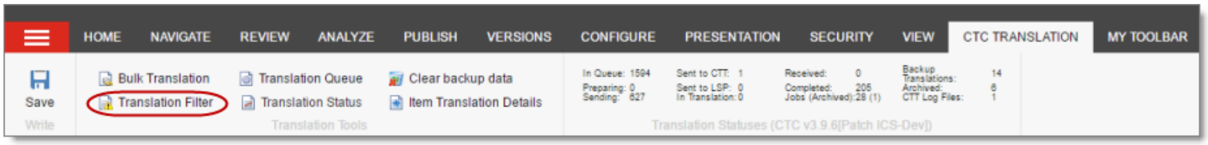
When a user sends out an item for translation, the Connector collects the content from all fields in the item with the field types specified in that parameter, in preparation for sending them out for translation. Then, if there are some fields that you do not want to send for translation, even though they belong to field types that the Connector sends out for translation, you can use the Translation Filter to exclude them.

1. Do one of the following to open the **Translation Filter** window:
   - In the Sitecore Content Editor ribbon, click the **CTC Translation** tab, and then in the **Translation Tools** section, click **Translation Filter**.
     - Sitecore 6 and 7



In Queue: 0 Sent to CTT: 3 Received: 0 Backup Translations: 137
Preparing: 0 Sent to LSP: 3 Completed: 221 Archived: 2
Sending: 0 In Translation: 269 Jobs (Archived): 66 (2) CTT Log Files: 8

     - Sitecore 8

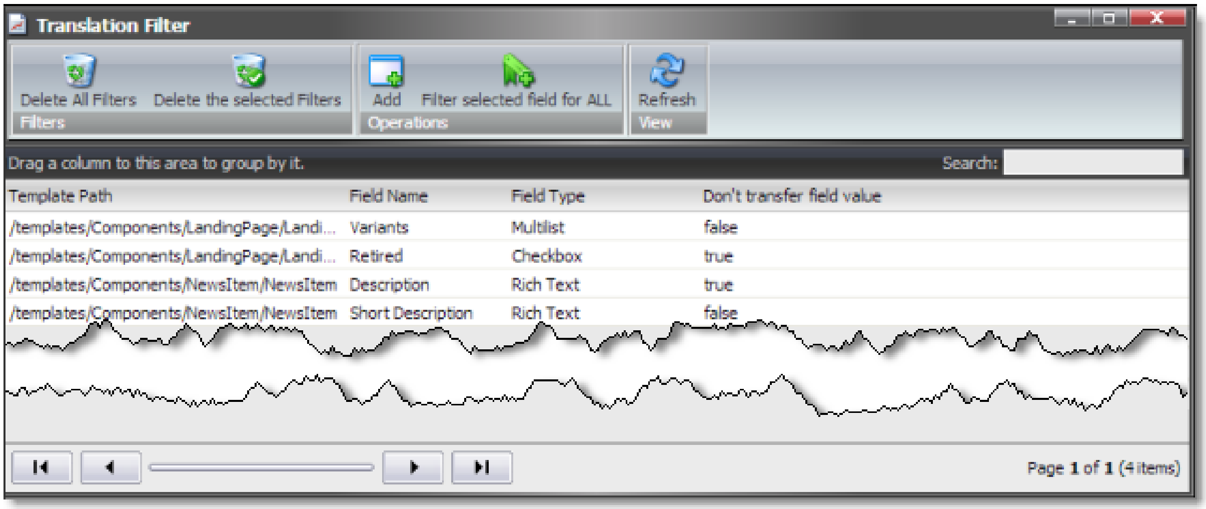


In Queue: 1594 Sent to CTT: 1 Received: 0 Backup Translations: 14
Preparing: 0 Sent to LSP: 0 Completed: 205 Archived: 6
Sending: 627 In Translation: 0 Jobs (Archived): 28 (1) CTT Log Files: 1

   - In the Sitecore Desktop, click the **Sitecore** menu, and then select **All Applications > CTC Translation > Translation Filter** from the menu.

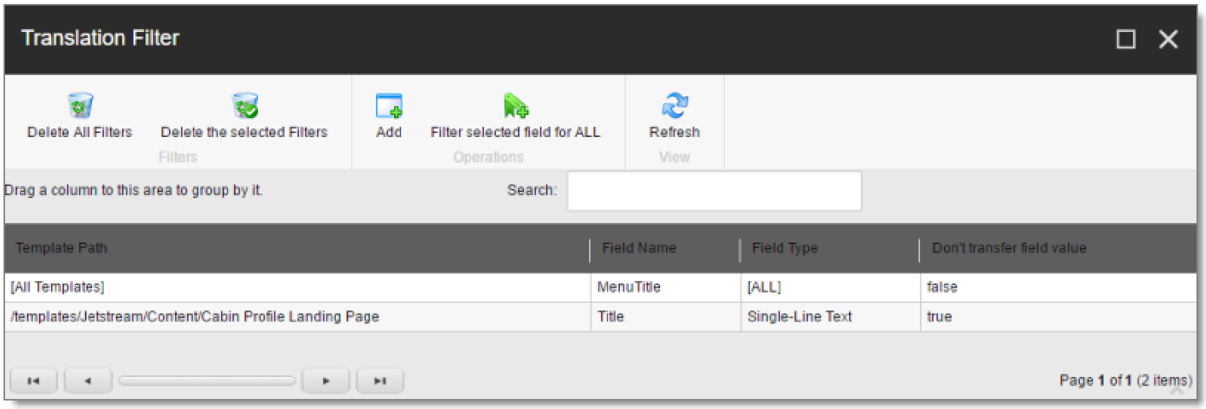
   The **Translation Filter** window opens.

   - If you use the Sitecore Content Editor in Sitecore 6 or 7, it opens in a separate browser window.
   - If you use the Sitecore Content Editor in Sitecore 8 or the Sitecore Desktop, it opens in the same browser window.
     - Sitecore 6 and 7


Translation Filter
Delete All Filters
Delete the selected Filters
Filters
Add
Filter selected field for ALL
Operations
Refresh
View
Drag a column to this area to group by it.
Search:
Page 1 of 1 (4 items)

Template Path	Field Name	Field Type	Don't transfer field value
/templates/Components/LandingPage/Landi...	Variants	Multilist	false
/templates/Components/LandingPage/Landi...	Retired	Checkbox	true
/templates/Components/NewsItem/NewsItem	Description	Rich Text	true
/templates/Components/NewsItem/NewsItem	Short Description	Rich Text	false

- Sitecore 8


Translation Filter
Delete All Filters
Delete the selected Filters
Filters
Add
Filter selected field for ALL
Operations
Refresh
View
Drag a column to this area to group by it.
Search:
Page 1 of 1 (2 items)

Template Path	Field Name	Field Type	Don't transfer field value
[All Templates]	MenuTitle	[ALL]	false
/templates/Jetstream/Content/Cabin Profile Landing Page	Title	Single-Line Text	true

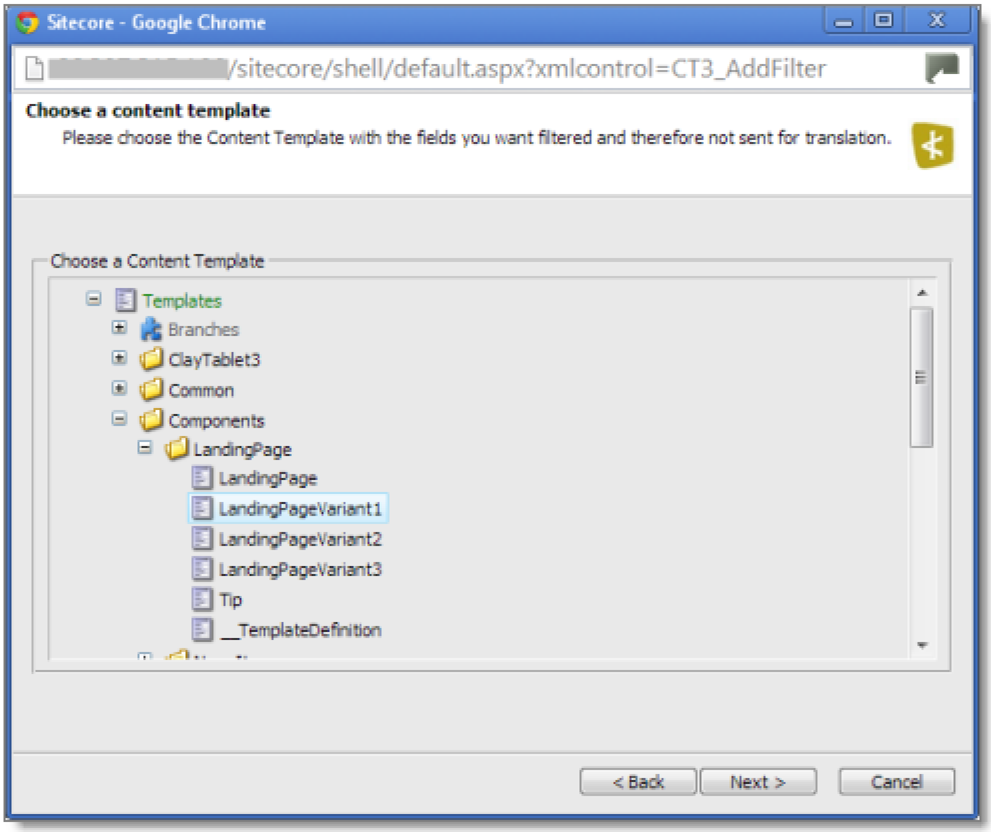
2. Click the **Add** button to add a new filter.

   The **Welcome** page of the **Field Filter** wizard opens.

3. Click **Next**.

   The **Choose a Content Template** page of the **Field Filter** wizard opens.

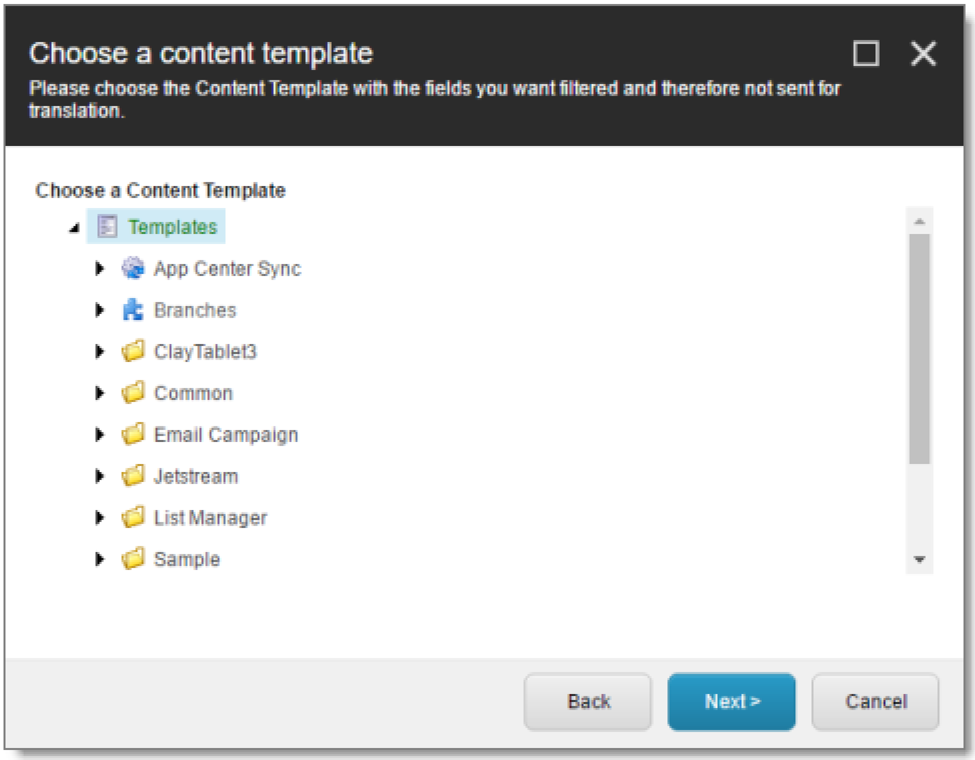
- Sitecore 6 and 7



Choose a content template

Please choose the Content Template with the fields you want filtered and therefore not sent for translation.

- Sitecore 8



Choose a content template

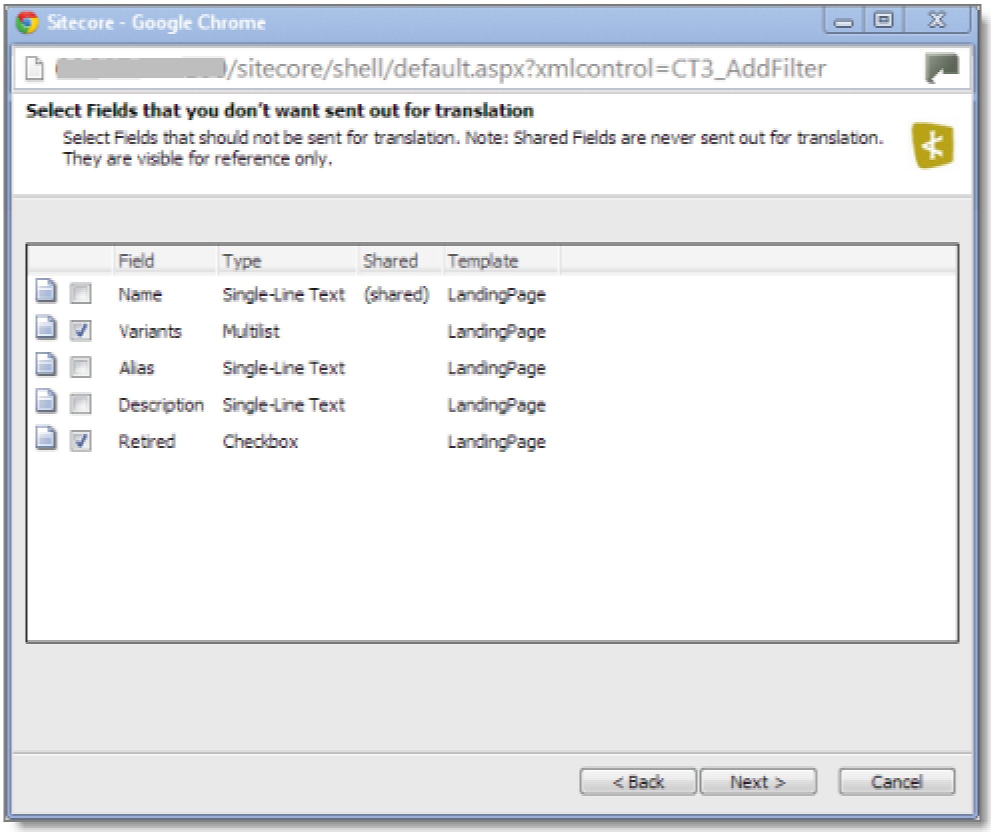
Please choose the Content Template with the fields you want filtered and therefore not sent for translation.

Choose a Content Template

4. Select the content template to associate with the fields you want to filter, and exclude from translation. Use the Expand ⊞ and Collapse ⊟ icons to navigate the tree. Click **Next**.

   The **Select Fields that you don't want sent out for translation** page of the **Field Filter** wizard opens.

   - Sitecore 6 and 7



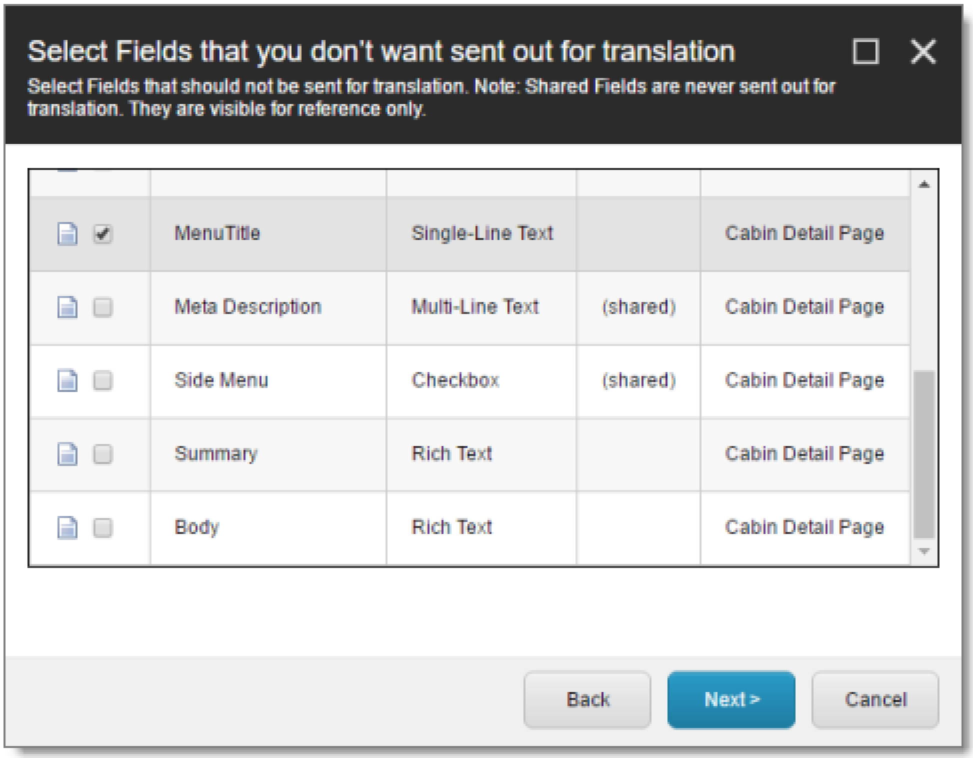
[redacted]/sitecore/shell/default.aspx?xmlcontrol=CT3_AddFilter

Select Fields that you don't want sent out for translation

Select Fields that should not be sent for translation. Note: Shared Fields are never sent out for translation. They are visible for reference only.

	Field	Type	Shared	Template
□	Name	Single-Line Text	(shared)	LandingPage
☑	Variants	Multilist		LandingPage
□	Alias	Single-Line Text		LandingPage
□	Description	Single-Line Text		LandingPage
☑	Retired	Checkbox		LandingPage

- Sitecore 8



Select Fields that you don't want sent out for translation

Select Fields that should not be sent for translation. Note: Shared Fields are never sent out for translation. They are visible for reference only.

☑	MenuTitle	Single-Line Text		Cabin Detail Page
□	Meta Description	Multi-Line Text	(shared)	Cabin Detail Page
□	Side Menu	Checkbox	(shared)	Cabin Detail Page
□	Summary	Rich Text		Cabin Detail Page
□	Body	Rich Text		Cabin Detail Page

This page displays the following information about all the fields in the template you selected:

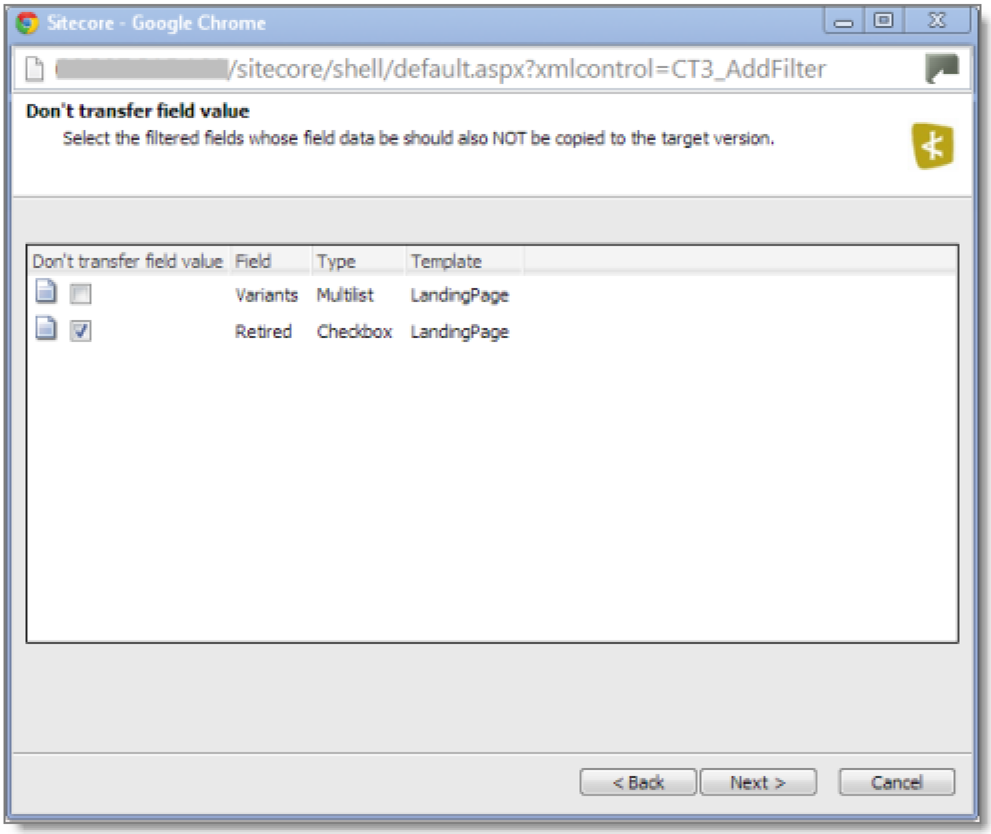
Column	Description
Field	The name of a field type in the template.
Type	The description of the field type.
Shared	Indicates whether the field type is shared.
Template	The template where the field type is located.

5. Select the check boxes of the field types you want to exclude from translation, and click **Next**.

Note: Shared fields are never sent for translation, so they are automatically excluded from translation. You do not need to select them.

The **Don't transfer field value** page of the **Field Filter** wizard opens.

- Sitecore 6 and 7



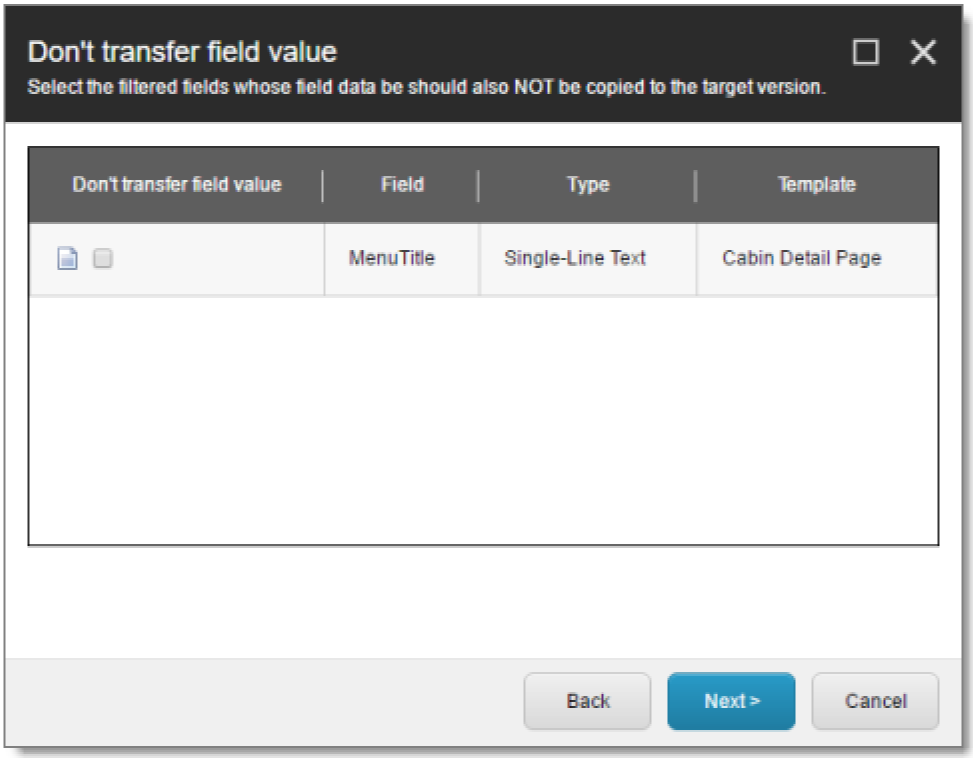
[redacted]/sitecore/shell/default.aspx?xmlcontrol=CT3_AddFilter

Don't transfer field value

Select the filtered fields whose field data be should also NOT be copied to the target version.

Don't transfer field value	Field	Type	Template
□	Variants	Multilist	LandingPage
☑	Retired	Checkbox	LandingPage

- Sitecore 8


Don't transfer field value
Select the filtered fields whose field data be should also NOT be copied to the target version.
Back
Next >
Cancel

Don't transfer field value	Field	Type	Template
□	MenuTitle	Single-Line Text	Cabin Detail Page

This page displays only the field types you selected in the previous page of the wizard.

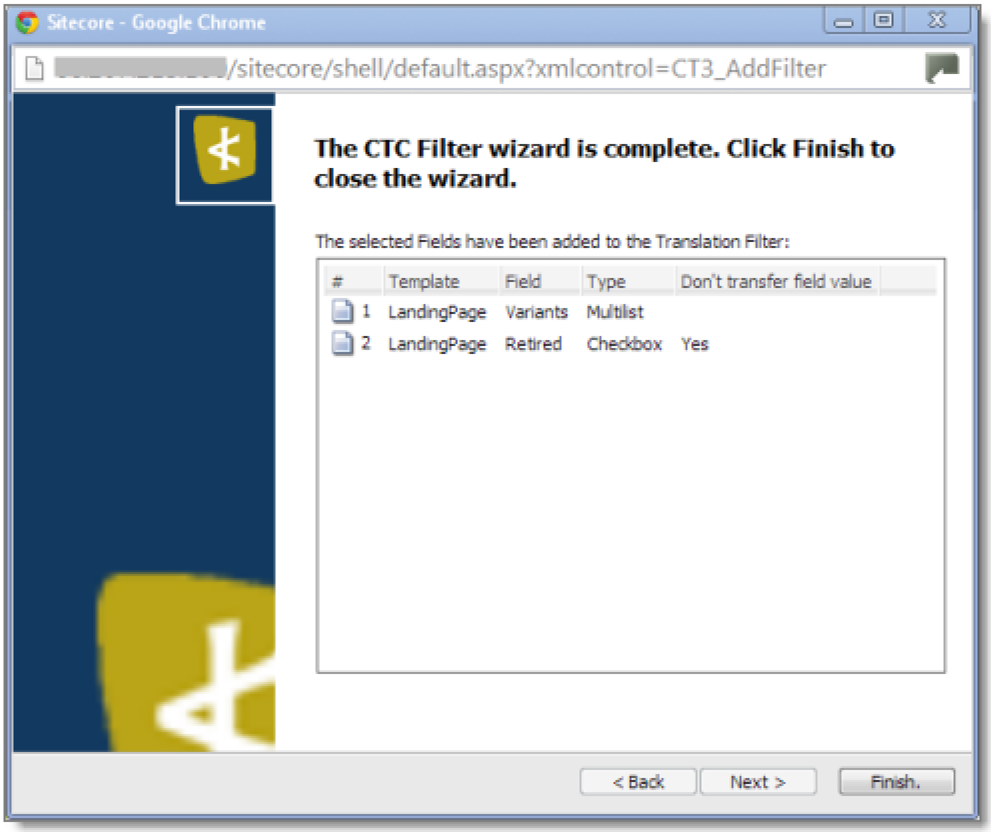
6. For each field type, indicate whether you want the Connector to copy the source-language content to the target version.

   - If you do not want the Connector to copy the source-language content of a field type to the target version, select the corresponding check box. This field type will be excluded from the target content.
   - If you want the Connector to copy the source-language content of a field type to the target version, clear the corresponding check box. This field type will be included in the target content; however, the content will be in the source language.

   Click **Next**.

   The summary page of the wizard opens.

   - Sitecore 6 and 7



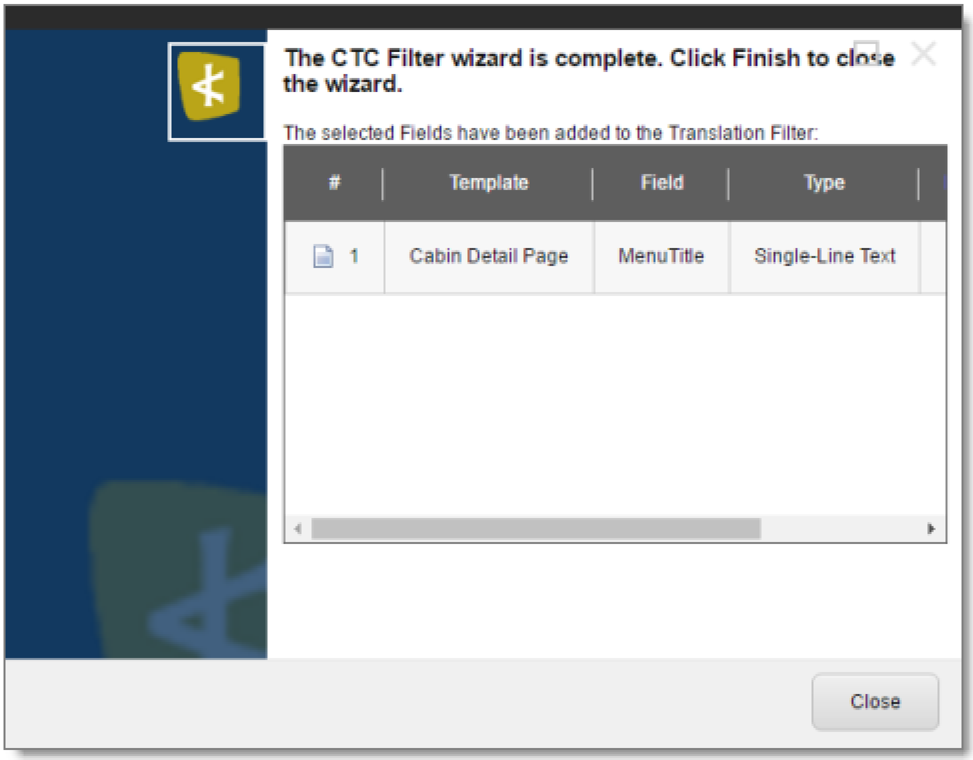
[redacted]/sitecore/shell/default.aspx?xmlcontrol=CT3_AddFilter

The CTC Filter wizard is complete. Click Finish to close the wizard.

The selected Fields have been added to the Translation Filter:

#	Template	Field	Type	Don't transfer field value
1	LandingPage	Variants	Multilist	
2	LandingPage	Retired	Checkbox	Yes

- Sitecore 8



The CTC Filter wizard is complete. Click Finish to close the wizard.

The selected Fields have been added to the Translation Filter:

#	Template	Field	Type
1	Cabin Detail Page	MenuTitle	Single-Line Text

This page displays the following information about the fields to exclude in the new filter.

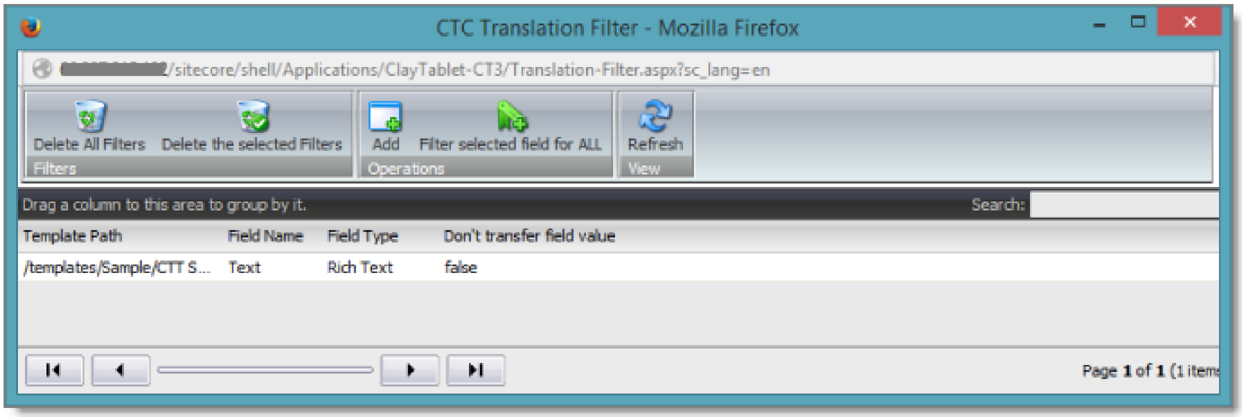
Column	Description
Template	The name of the template where the field type to exclude from translation is located.
Field	The name of the field type to exclude from translation.
Type	The type of field to exclude from translation.
Don't transfer field value	Indicates whether the Connector copies the source-language content to the target version for this field type. ■ Blank indicates that the Connector *does* copy the source-language content of a field to the target version, select the corresponding check box. This field will be included in the target content; however, it will be in the source language. ■ `yes` indicates that the Connector *does not* copy the source-language content of a field type to the target version. The Connector does not automatically transfer the value of the field in the source version to the field in the target version.

7. Click **Finish**.

8. Sitecore 8 only. A message box opens, confirming that you want to close the wizard. Click **OK**.

   The wizard closes, and the **CTC Translation Filter** window displays each filter you created in a separate row:

   - Sitecore 6 and 7


CTC Translation Filter - Mozilla Firefox
Delete All Filters
Delete the selected Filters
Filters
Add
Filter selected field for ALL
Operations
Refresh
View
Search:

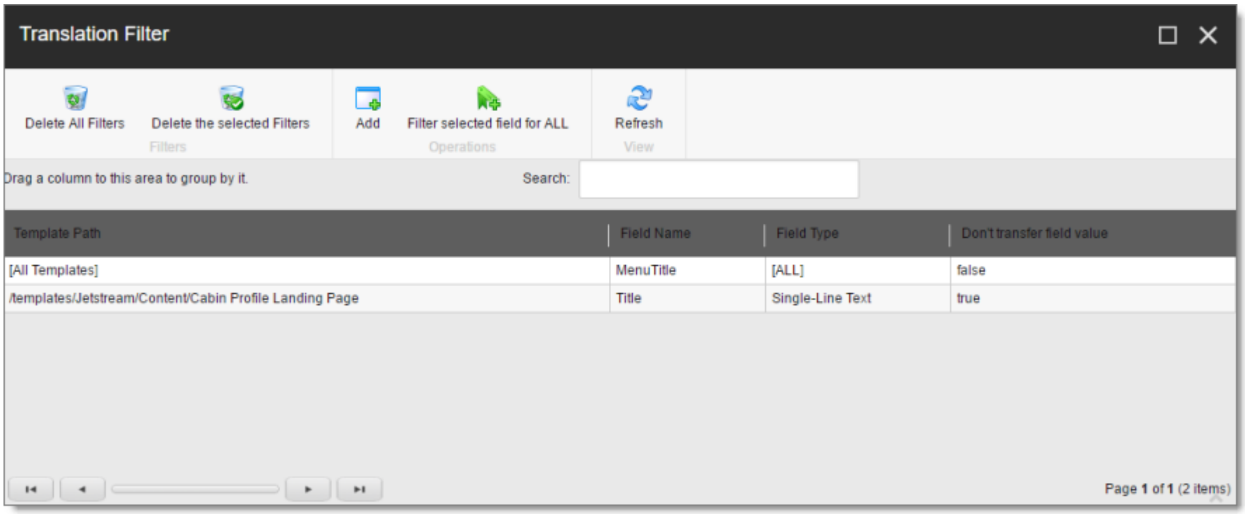
[redacted]/sitecore/shell/Applications/ClayTablet-CT3/Translation-Filter.aspx?sc_lang=en

Drag a column to this area to group by it.

Template Path	Field Name	Field Type	Don't transfer field value
/templates/Sample/CTT S...	Text	Rich Text	false

Page 1 of 1 (1 item

- Sitecore 8



Drag a column to this area to group by it.

Template Path	Field Name	Field Type	Don't transfer field value
[All Templates]	MenuTitle	[ALL]	false
/templates/Jetstream/Content/Cabin Profile Landing Page	Title	Single-Line Text	true

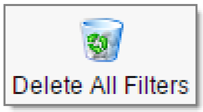
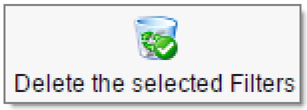
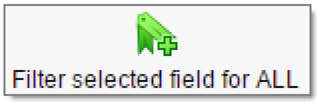
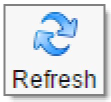
Page 1 of 1 (2 items)

This window displays the following information about the filters:

Column	Description
Template Path	The path and name of the template. Field types in this template can be excluded from translations.
Field Name	When the Connector sends out content items based the specified template for translation, it does not send out this field's content for translation.
Field Type	The type of this field to exclude from translation. When `[ALL]` is displayed, then all types of this field are excluded from translation.

Column	Description
Don't transfer field value	Indicates whether the Connector copies the source-language content to the target version for this field type. - `false` indicates that the Connector *does* copy the source-language content of a field type to the target version, select the corresponding check box. This field will be included in the target content, however it will be in the source language. - `true` indicates that the Connector *does not* copy the source-language content of a field to the target version. The Connector does not automatically transfer the value of the field in the source version to the field in the target version.

Now that you have created a filter, you can perform the following actions in the **CTC Translation Filter** window (in addition to the **Add Filter** button, which was described above in step 3):

Button	Description
	Deletes all filters displayed in the CTC Translation Filter window.
	Deletes all selected filters from the CTC Translation Filter window.
	Applies the selected filter to all templates with the same field type.
	Refreshes the list of displayed filters.

4.10 Configuring Connector Roles and Adding Users

The Connector installs the roles described below into Sitecore:

Role	Description
sitecore\Translation Administrator	Users added to this role can set translation filters, remove completed job status, and delete the Connector backup data. Users can submit content to the translation queue or send it out immediately to translation.

Role	Description
sitecore\Translation Operator	Users added to this role can submit content to the translation queue. They can also send items from the queue for translation and remove items from the queue. When using the Bulk Translation feature, these users can skip the translation queue. They cannot send out individual content items directly to translation.
sitecore\ Translation ItemSent Notification Receiver	Users added to this role receive email notifications from the Connector when it sends items out for translation.
sitecore\ Translation ItemCompleted Notification Receiver	Users added to this role receive email notifications from the Connector when it receives translated items back from translation.

You must assign your users to these roles so they can access the functionality described above. For detailed instructions, refer to the *Sitecore CMS Security Administrator's Cookbook*. This guide is available in the documentation section of the Sitecore Developer Network (SDN) site, at http://sdn.sitecore.net/Reference.aspx. You must log in to this site to access the documentation.

4.11 Configuring the In-Context Preview Feature

You can configure the settings for the In-Context Preview feature. This feature enables those without Sitecore access (such as translators) to preview items in the XML files for translation as Sitecore would render them.

Warning: This feature is currently in alpha testing. It is not currently generally available for production usage. Do not configure this feature unless you have contacted Clay Tablet and arranged to be an alpha tester of this feature. If you are not an alpha tester of this feature, leave all these settings blank.

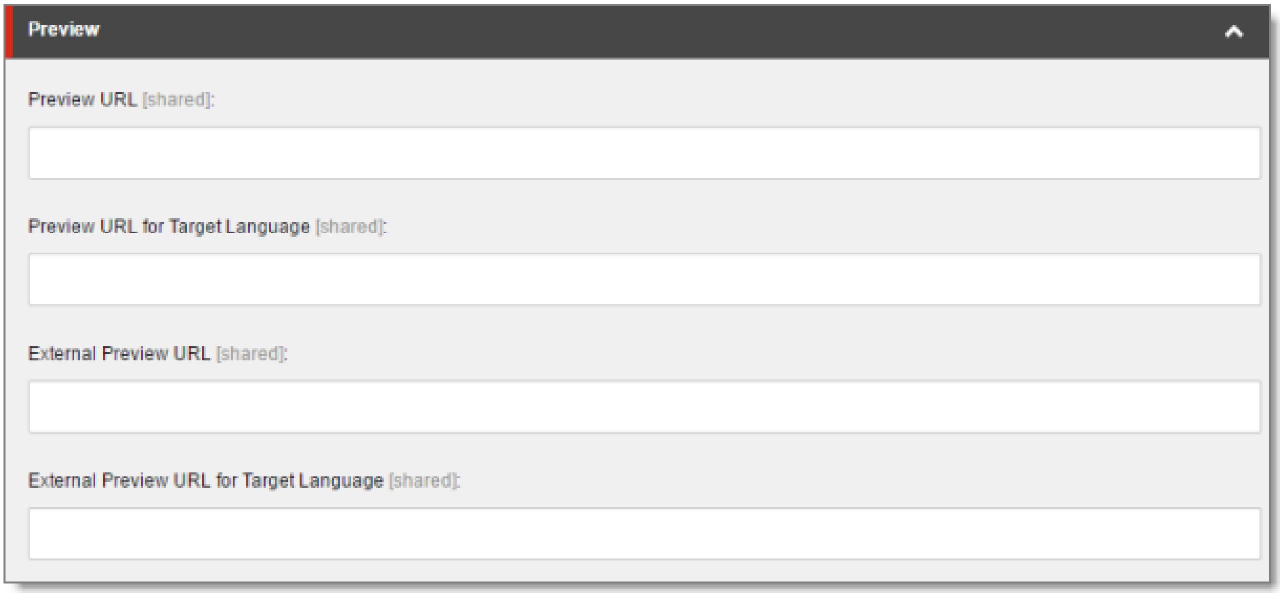
To configure the In-Context Preview feature:

1. In the Content Editor, in the content tree, navigate to the `/Sitecore/System/Settings/Clay Tablet Settings` folder.

   Tip: Scroll down in the `Settings` folder to locate the `Clay Tablet Settings` sub-folder.

2. Click the **Clay Tablet Settings** item in the folder to open it in the Content Editor.

3. Scroll down to the **Preview** section.



Preview

Preview URL [shared]:

Preview URL for Target Language [shared]:

External Preview URL [shared]:

External Preview URL for Target Language [shared]:

4. Specify the following settings:

Tip: You can use the parameters listed in the table below this table to configure the URLs.

Setting	Description	Sample Value
Preview URL	The URL that the Clay Tablet Preview server uses to access the Sitecore instance and preview the source version of the item sent to translation. Typically this server needs access through your firewall to access the URL and credentials to log into your Sitecore instance.	http://yoursitecoreurl/?sc_itemid={id}&sc_mode=preview&sc_lang={lang}
Preview URL for Target Language	The URL that the Clay Tablet Preview server uses to access the Sitecore instance and preview the target version of the item sent to translation. Typically this server needs access through your firewall to access the URL and credential to log into your Sitecore instance.	http://yoursitecoreurl/?sc_itemid={id}&sc_mode=preview&sc_lang={targetlang}
External Preview URL	The URL that enables translators or external reviewers to preview the source item on the Clay Tablet Preview server. For instructions on configuring this setting, contact Clay Tablet Support, as described in "How to Contact Clay Tablet Support" on page 9.	https://ct-preview-server-url?chunk={chunkid}&lang={lang}

Setting	Description	Sample Value

External Preview URL for Target Language	The URL that enables translators or external reviewers to preview the target item the Clay Tablet Preview server. For instructions on configuring this setting, contact Clay Tablet Support, as described in "How to Contact Clay Tablet Support" on page 9.	https://ct-preview-server-url?chunk={chunkid}
		&lang={targetlang}

You can use the following parameters listed in the table below to configure the URLs described above:

Parameter	Description
{id}	The identifier of the source Sitecore item sent for translation.
{version}	The version of the source Sitecore item sent for translation.
{lang}	The source language of the Sitecore item sent for translation.
{targetversion}	The version of the translated Sitecore item.
{targetlang}	The target language of the translated Sitecore item.
{jobid}	The Connector translation job ID.
{chunkid}	The globally unique ID generated by the Connector, which identifies each translatable field (chunk) in the XML file generated by Clay Tablet. **Note:** This parameter is relevant only to the **External Preview URL** and **External Preview URL for Target Language** settings, described above.

5 Installing the Enhanced Workbox

Clay Tablet provides an optional enhanced Workbox, which has more features than the standard Sitecore Workbox. Installing Clay Tablet's enhanced Workbox updates and replaces the original Sitecore Workbox.

This Workbox supports much more powerful management of large lists of items in a workflow. You can sort by different column headings, approve many items at once and manage the display of long lists easily.

Important: Install Clay Tablet's enhanced Workbox only if you are comfortable with replacing the standard Sitecore Workbox.

To install the enhanced Workbox:

1. On the Sitecore Desktop, on the Windows Start menu, select **Sitecore > Development Tools > Installation Wizard**.

   The **Welcome** page of the **Sitecore Install Package** wizard opens.

2. Click **Next**.

   The **Select Package** page of the **Sitecore Install Package** wizard opens.

3. Click **Browse**, and locate the Workbox installation package, `CT_Sitecore_WorkBox_update-x.y.z.zip`, where *x.y.z* is the current version number of the Connector for Sitecore.

   Tip: The installation package is in the following location in the delivery package: `<Delivery Package/Workbox/CT_Sitecore_WorkBox_update-x.y.z.zip>`.

4. Click **Upload** to upload the package.

5. After the package is uploaded, click **Next**.

6. Follow any on-screen instructions to finish installing the Workbox package into Sitecore.

7. Do one of the following:

   - **Sitecore 6.4 and higher, Sitecore 7, and Sitecore 8.0:** On the Sitecore server, navigate to the `web.config` file and open it in a text editor. For example, the location of this file may be `C:\inetpub\wwwroot\Sitecore66\Website`.
   - **Sitecore 8.1:** On the Sitecore server, navigate to the `sitecore.config` file and open it in a text editor. For example, the location of this file may be `C:\inetpub\wwwroot\Sitecore81\Website\App_Config`.

8. Search for `workflowProvider`.

   Note: There are two places where `workflowProvider` is configured, one for the `core` database (in the section starting with `<!-- core -->`), and the other for the `master` database (in the section starting with `<!-- master -->`),. You change only the `workflowProvider` section for the `master` database.

9. Comment out the `workflowProvider` section for the `master` database.

10. Add the following section instead:

```
<workflowProvider hint="defer"
type="Sc.ClayTablet.WorkboxEx.SortingWorkflowProvider, Sc.ClayTablet.WorkboxEx">

  <param desc="database">$(id)</param>
  <param desc="history store" ref="workflowHistoryStores/main" param1="$(id)"/>
</workflowProvider>
```

11. Verify that after editing, the relevant section in the `web.config` or `sitecore.config` file is similar to the following:

```
<database id="master" singleInstance="true" type="Sitecore.Data.Database,
Sitecore.Kernel">
......
  <!--
  <workflowProvider hint="defer
            type="Sitecore.Workflows.Simple.WorkflowProvider,
            Sitecore.Kernel">
    <param desc="database">$(id)</param>
    <param desc="history store" ref="workflowHistoryStores/main" param1="$(id)
                "/>

  </workflowProvider>
  -->

  <workflowProvider hint="defer"
            type="Sc.ClayTablet.WorkboxEx.SortingWorkflowProvider,
    Sc.ClayTablet.WorkboxEx">
    <param desc="database">$(id)</param>
    <param desc="history store" ref="workflowHistoryStores/main" param1="$(id)
    "/>
  </workflowProvider>
  .......
```

12. Save the `web.config`or `sitecore.config` file.

Sitecore restarts.

For information on using the enhanced Workbox, refer to the *Clay Tablet Connector for Sitecore User Guide*.

6 Pre-Production Testing

After you complete the configuration, your Clay Tablet Connector for Sitecore installation is ready for testing. We recommend sending only a few pages for translation in one language as an initial test. For detailed instructions, refer to the *Clay Tablet Connector for Sitecore User Guide*. Once successful, you can send as many languages as required.

Please coordinate with your translation provider for this test process.

If you have any concerns or questions, please contact Clay Tablet Support. For details, see "How to Contact Clay Tablet Support" on page 9.

7 Appendix: Connector Translation File Formats

7.1 XML Translation File – New Translation

The following is an example of new content that the Connector sends for translation:

```
<TranslationContent CT2ProjectId="5ab2e78e-e427-4b89-9629-562c13584d94"
CT2AssetId="9250c3ee-da56-47d8-880e-2514052d7a58" CT2SourceLanguageCode="en-US"
CT2TargetLanguageCode="fr-FR" SitecoreSourceLanguageCode="en"
SitecoreTargetLanguageCode="fr-FR">

  <SitecoreItem DatabaseName="master" ItemId="{E6F3AA2D-3E5B-47A5-8B4A-
  1EA603C9A3FA}" ItemSourceVersion="1" ItemTargetVersion="1"
  TranslationDeadline="">

    <FieldContent FieldName="Title">Demo content to translate</FieldContent>

    <FieldContent FieldName="Body">

      &lt;p&gt;This is content that google will try to translate&lt;/p&gt;

    </FieldContent>

  </SitecoreItem>

</TranslationContent>
```

7.2 HTML Translation File – New Translation with Metadata in Comments

The following is an example of new content that the Connector sends for translation, with metadata in the comments:

```
<!--CT2Translation:[From:en-US To:fr-FR]-->

<!--CT2SharedMeta:92d2c042-2bcf-41cd-8a04-fe4d327f2157|45525ffe-fdf1-403f-be6a-
43bcd99e3373|en|fr-FR|en-US|fr-FR:SharedMetaEnd-->

<!--CT2FieldMeta:master|{C069745D-63F6-4EB1-868F-EF1AE7C0BB76}|2|10|Title:Begin-->

Title of demo2

<!--CT2MetaField::End-->

<!--CT2FieldMeta:master|{C069745D-63F6-4EB1-868F-EF1AE7C0BB76}|2|10|Body:Begin-->

<p> Body of demo2 </p>

<!--CT2MetaField::End-->
```

7.3 XML Translation File – Correction Requested

The following is an example of a request to correct the translation:

```
<TranslationCorrectionContent CT2ProjectId="3149ca06-64f8-4e07-afb1-57a7c3539670"
CT2AssetId="08a44fe1-a650-4dc7-bca5-c63e6cd0a4f7" CT2SourceLanguageCode="en-US"
CT2TargetLanguageCode="el-GR" SitecoreSourceLanguageCode="en"
SitecoreTargetLanguageCode="el-GR" DatabaseName="master" ItemId="{6C95A5D0-30D1-
4329-95BD-E9C8FF71FD07}" ItemSourceVersion="1" ItemTargetVersion="4">
  <Notes>
  </Notes>
  <ItemFields>
    <ItemField FieldName="Title">
      <SourceContent>Smartphone traffic is up 193% in a year</SourceContent>
      <TargetContent>Smartphone κίνηση είναι επάνω 193% σε ένα
      χρόνο</TargetContent>
    </ItemField>
  </ItemFields>
</TranslationCorrectionContent>
```

7.4 HTML Translation File – Correction Requested with Metadata in Comments

The following is an example of a request to correct the translation with metadata in the comments:

```
<!--CT2TranslationCorrection:[From:en-US To:de-DE]-->
<!--CT2SharedMeta:95e71b0c-5e75-4a34-8cf7-e706d7edff2d|f366b6e9-78c9-438a-b67a-
37a938257889|en|de-DE|en-US|de-DE:SharedMetaEnd-->
<!--CT2ItemMeta:master|{333E5B70-BA26-402B-A3CE-5A5FBAFF7D66}|4|9:ItemMetaEnd-->
<!--CT2Note::Begin-->
The translation of the title needs correction
<!--CT2Note::End-->
<!--CT2FieldMeta:Title:Begin-->
<!--Source Content::Begin-->
Smartphone traffic is up 193% in a year
<!--Source Content::End-->
<!--Translated Content::Begin-->
Smartphone-Verkehr ist bis 193% in einem Jahr
<!--Translated Content::End-->
<!--CT2MetaField::End-->
```

7.5 Translation Memory Update File

In the following example, a user corrected the translation and it must be sent back to the translation provider to update the remote translation memory (TM):

```
<UpdateTMContent CT2SourceLanguageCode="en-US" CT2TargetLanguageCode="da-DK"
SitecoreSourceLanguageCode="en" SitecoreTargetLanguageCode="da-DK">
  <ContentCorrections>
    <ContentCorrection>
      <SourceContent>Smartphone traffic is up 193% in a year</SourceContent>
      <TargetContent>Smartphone trafik er op 193% på et år</TargetContent>
    </ContentCorrection>
    <ContentCorrection>
      <SourceContent>
        &lt;p&gt;Smartphone traffic is up. Feature phone share is down. And
        traffic from mobile Internet devices (like the iPod touch) that don't
        have built-in phones is booming — even before Apple releases the
        iPad.&lt;/p&gt;
        &lt;p&gt;That's the thrust of the the latest report by AdMob, the mobile
        advertising network snapped up last fall by Google (&lt;a
        href="http://money.cnn.com/quote/quote.html?symb=GOOG"
        rel="external"&gt;GOOG&lt;/a&gt;) before Apple (&lt;a
        href="http://money.cnn.com/quote/quote.html?symb=AAPL"
        rel="external"&gt;AAPL&lt;/a&gt;) could buy it.&lt;/p&gt;</SourceContent>
      <TargetContent>&lt;p&gt; Smartphone trafik er op. Feature telefon andel er
      nede. Og trafik fra det mobile internet-enheder (ligesom iPod touch), der
      ikke har indbygget telefoner boomer - selv før Apple frigiver IPAD.
      &lt;/p&gt;&lt;p&gt; Det er essensen af den seneste rapport fra admob, det
      mobile annoncenetværk snapped op sidste efterår af Google ( &lt;a
      href="http://money.cnn.com/quote/quote.html?symb=GOOG"
      rel="external"&gt;GOOG&lt;/a&gt; ) før Apple ( &lt;a
      href="http://money.cnn.com/quote/quote.html?symb=AAPL"
      rel="external"&gt;AAPL&lt;/a&gt; ) kunne købe det.
      &lt;/p&gt;</TargetContent>
    </ContentCorrection>
  </ContentCorrections>
</UpdateTMContent>
```

8 Appendix: Language Codes

For detailed instructions on setting up the correct Connector language codes for every Sitecore language your company uses for translation, see "Configuring the Sitecore Languages in the Sitecore Content Editor" on page 41.

The Clay Tablet Connector has the following language codes:

Language Identifier	Language Code
Afrikaans	"af-ZA"
Albanian	"sq-AL"
Amharic	"am-ET"
Arabic_Algeria	"ar-DZ"
Arabic_Bahrain	"ar-BH"
Arabic_Egypt	"ar-EG"
Arabic_Iraq	"ar-IQ"
Arabic_Jordan	"ar-JO"
Arabic_Kuwait	"ar-KW"
Arabic_Lebanon	"ar-LB"
Arabic_Libya	"ar-LY"
Arabic_MiddleEast	"ar-XR"
Arabic_Morocco	"ar-MA"
Arabic_Oman	"ar-OM"
Arabic_Qatar	"ar-QA"
Arabic_Saudi_Arabia	"ar-SA"
Arabic_Syria	"ar-SY"
Arabic_Tunisia	"ar-TM"
Arabic_UAE	"ar-AE"
Arabic_Yemen	"ar-YE"

Language Identifier	Language Code
Armenian	"hy-AM"
Assamese	"as-IN"
Basque	"eu-ES"
Belarusian	"be-BY"
Bengali_Bangladesh	"bn-BD"
Bengali_India	"bn-IN"
Bosnian_Bosnia_Herzegovina	"bs-BA"
Bulgarian	"bg-BG"
Burmese	"my-MM"
Catalan	"ca-ES"
Chinese_Hong_Kong	"zh-HK"
Chinese_Macao	"zh-MO"
Chinese_PRC	"zh-CN"
Chinese_Singapore	"zh-SG"
Chinese_Taiwan	"zh-TW"
Croatian	"hr-HR"
Croatian_Bosnia_Herzegovina	"hr-BA"
Czech	"cs-CZ"
Danish	"da-DK"
Divehi	"dv-MV"
Dutch	"nl-NL"
Dutch_Belgium	"nl-BE"
English_Australia	"en-AU"

Language Identifier	Language Code
English_Belize	"en-BZ"
English_Canada	"en-CA"
English_HongKong	"en-HK"
English_India	"en-IN"
English_Indonesia	"en-ID"
English_Ireland	"en-IE"
English_Jamaica	"en-JM"
English_Malaysia	"en-MY"
English_New_Zealand	"en-NZ"
English_Philippines	"en-PH"
English_Singapore	"en-SG"
English_South_Africa	"en-ZA"
English_Trinidad	"en-TT"
English_UK	"en-GB"
English_US	"en-US"
English_Zimbabwe	"en-ZW"
Estonian	"et-EE"
Faroese	"fo-FO"
Farsi	"fa-IR"
Filipino	"fil-PH"
Finnish	"fi-FI"
French	"fr-FR"
French_Belgium	"fr-BE"

Language Identifier	Language Code
French_Cameroon	"fr-CM"
French_Canada	"fr-CA"
French_Cote_d_Ivoire	"fr-CI"
French_Democratic_Rep_Congo	"fr-CD"
French_Haiti	"fr-HT"
French_Luxembourg	"fr-LU"
French_Mali	"fr-ML"
French_Monaco	"fr-MC"
French_Morocco	"fr-MA"
French_Reunion	"fr-RE"
French_Senegal	"fr-SN"
French_Switzerland	"fr-CH"
Frisian_Netherlands	"fy-NK"
Fulfulde_Nigeria	"ff-NG"
FYRO_Macedonian	"mk-MK"
Gaelic_Ireland	"gd-IE"
Gaelic_Scotland	"gd-GB"
Gallegan	"gl-ES"
Georgian	"ka-GE"
German	"de-DE"
German_Austria	"de-AT"
German_Liechtenstein	"de-LI"
German_Luxembourg	"de-LU"

Language Identifier	Language Code
German_Switzerland	"de-CH"
Greek	"el-GR"
Guarani	"gn-PY"
Gujarati	"gu-IN"
Hausa	"ha-NE"
Hawaiian	"haw-US"
Hebrew	"he-IL"
Hindi	"hi-IN"
Hungarian	"hu-HU"
Icelandic	"is-IS"
Igbo	"ig-NG"
Indonesian	"id-ID"
Inuktitut	"iu-CA"
Italian	"it-IT"
Italian_Switzerland	"it-CH"
Japanese	"ja-JP"
Kannada	"kn-IN"
Kanuri	"kr-TD"
Kashmiri	"ks-IN"
Kazakh	"kk-KZ"
Khmer	"km-KH"
Konkani	"kok-IN"
Korean	"ko-KR"

Language Identifier	Language Code
Kyrgyz	"ky-KZ"
Lao	"lo-LA"
Latin	"la-XL"
Latvian	"lv-LV"
Lithuanian	"lt-LT"
Malay	"ms-MY"
Malay_Brunei_Darussalam	"ms-BN"
Malayalam	"ml-IN"
Maltese	"mt-MT"
Maori	"mi-NZ"
Marathi	"mr-IN"
Mongolian	"mn-MN"
Nepali	"ne-NP"
Nepali_India	"ne-IN"
Norwegian	"nb-NO"
Norwegian_Nynorsk	"nn-NO"
Oriya	"or-IN"
Oromo	"om-ET"
Panjabi	"pa-PK"
Polish	"pl-PL"
Portuguese	"pt-PT"
Portuguese_Brazil	"pt-BR"
Punjabi_Pakistan	"pa-PK"

Language Identifier	Language Code
Pushto	"ps-AF"
Quechua_Ecuador	"qu-EC"
Quechua_Peru	"qu-PE"
Rhaeto_Romance	"rm-IT"
Romanian	"ro-RO"
Romanian_Moldova	"ro-MD"
Russian	"ru-RU"
Russian_Moldava	"ru-MD"
Sami	"se-NO"
Sanskrit	"sa-IN"
Serbian_Cyrillic	"sr-RS"
Serbian_Latin	"sr-SP"
Sindhi_India	"sd-IN"
Sindhi_Pakistan	"sd-PK"
Sinhala	"si-LK"
Slovak	"sk-SK"
Slovenian	"sl-SI"
Somali	"so-ET"
Sorbian	"wen-DE"
Spanish	"es-ES"
Spanish_Argentina	"es-AR"
Spanish_Bolivia	"es-BO"
Spanish_Chile	"es-CL"

Language Identifier	Language Code
Spanish_Colombia	"es-CO"
Spanish_Costa_Rica	"es-CR"
Spanish_Dominican_Republic	"es-DO"
Spanish_Ecuador	"es-EC"
Spanish_El_Salvador	"es-SV"
Spanish_Honduras	"es-HN"
Spanish_LatinAmerica	"es-XL"
Spanish_Mexico	"es-MX"
Spanish_Nicaragua	"es-NI"
Spanish_Panama	"es-PA"
Spanish_Paraguay	"es-PY"
Spanish_Peru	"es-PE"
Spanish_Puerto_Rico	"es-PR"
Spanish_Uruguay	"es-UY"
Spanish_US	"es-US"
Spanish_Venezuela	"es-VE"
Swahili	"sw-TZ"
Swedish	"sv-SE"
Swedish_Finland	"sv-FI"
Syriac	"syr-SY"
Tajik	"tg-TJ"
Tamil	"ta-IN"
Tatar	"tt-RU"

Language Identifier	Language Code
Telugu	"te-IN"
Thai	"th-TH"
Tibetan	"bo-CN"
Tigrinya_Eritrea	"ti-ER"
Tigrinya_Ethiopia	"ti-ET"
Tsonga	"ts-ZA"
Tswana	"tn-BW"
Turkish	"tr-TR"
Turkmen	"tk-TM"
Uighur	"ug-CN"
Ukrainian	"uk-UA"
Urdu	"ur-PK"
Urdu_India	"ur-IN"
Uzbek	"uz-UZ"
Venda	"ve-ZA"
Vietnamese	"vi-VN"
Welsh	"cy-GB"
Xhosa	"xh-ZA"
Yi	"ii-CN"
Yiddish	"yi-MD"
Yoruba	"yo-NG"
Zulu	"zu-ZA"

9 Appendix: Connector Extensions

You can create the following extensions to the Connector:

- "Configuring the Behavior of the Send Dependent Items check box in the Bulk Translation Wizard" on page 81
- "Using Custom Logic to Update a Remote TM" on page 82
- "Calling Custom Workflow Logic" on page 85
- "Using Custom Logic to Overwrite the Workflow Target-Language Setting when Automatically Sending Items to the Translation Queue via Workflow" on page 87
- "Using Custom Logic to Configure how the UploadService Creates Jobs when Automatically Sending Out Items from the Translation Queue" on page 88
- "Using Custom Logic to Populate the Translation Queue" on page 91

9.1 Configuring the Behavior of the Send Dependent Items check box in the Bulk Translation Wizard

By default, if the **Send Dependent Items** check box is selected in the **Root of Source Item** page of the **Bulk Translation** wizard, the Connector sends out only data source items of the item itself.

Alternatively, you can develop your own pipeline to determine which dependent items the Connector sends for translation when this check box is selected. You configure this pipeline in the `<pipelines>` section of the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file. If the **Send Dependent Items** check box is selected and the pipeline is defined, then when you run the Bulk Translation wizard:

1. The Connector calls the pipeline for each item selected in the **Bulk Translation** wizard.
2. On its own, the pipeline finds dependent items recursively.

   Important: The pipeline overrides the default Connector behavior of sending out only data source items of the item itself. The Connector does not call the pipeline again using a dependent item as a source for recursively searching for additional dependent items.

3. The Connector adds the results to the list of items to send out for translation.

To develop a pipeline:

1. Insert the logic to recursively determine all dependent items of an item to send out for translation, similar to the following example:

```
<CustomRecursiveDependentItemsPipeline>
<!-- Insert the logic to recursively determine all dependent items of an item to
send out for translation.
-->
```

```
<processor type="Your.Class, Your.Assembly" method="getDependentItems"/>
</CustomRecursiveDependentItemsPipeline>
```

2. Ensure that the processor method has the following signature:

```
public void getDependentItems(ClayTablet.SC.Pipelines.DependentItemsPipelineArgs
args);
```

3. Use the args parameter to retrieve information about a source item. You will search for dependent items of this source:

```
public class DependentItemsPipelineArgs {
String ItemDatabase { get; }
String ItemId { get; }
String SourceLanguage { get; }
String SourceVersion { get; }
String[] TargetLanguages { get; }
List<String> DependentItems { get; }
}
```

4. Insert the item IDs of all dependent items of the source item into the DependentItems that the pipeline returns.

9.2 Using Custom Logic to Update a Remote TM

You can call the <SendTmUpdatePipeline> pipeline to send out updates to a remote TM (translation memory) of selected items and versions for which the translated versions were updated.

To call this pipeline:

1. Create an instance of ClayTablet.SC.Pipelines.SendTmUpdatePipelineArgs class to pass to the pipeline as an argument. The SendTmUpdatePipelineArgs class is defined in ClayTablet.SC.dll as:

```
public class SendTmUpdatePipelineArgs : PipelineArgs
{
  public String SourceLanguage { get; set; }
  public String TargetLanguage { get; set; }
  public void AddTmUpdateItem(string itemId, int sourceVersion, int
  targetVersion);
  public List<TmUpdateItem> TmUpdateItems { get; }
}
```

2. In SendTmUpdatePipelineArgs, specify the following information:

Information	Description
ItemId	The Sitecore item ID of each updated item to send to the remote TM.
SourceLanguage	The source language of the updated items to send to the remote TM.
TargetLanguage	The target language of the updated items to send to the remote TM.
AddTmUpdateItem	Call this one or more times to add a set of `ItemId`/`SourceVersion`/`TargetVersion` to the list.
SourceVersion	The source-language version of the updated items to send to the remote TM.
TargetVersion	The target-language version of the updated items to send to the remote TM.

Note: Invoking the pipeline once can update the remote TM for multiple Sitecore items with the same source and target languages.

3. Invoke the pipeline to send the the specified Sitecore items with the same source and target languages to update the remote TM.

4. Use the `SendTmUpdatePipelineArgs.TmUpdateItems` class to verify that an item has actually been sent out to update a remote TM. class is defined in `ClayTablet.SC.dll` as:

```
public class TmUpdateItem
{
  public string ItemId { get; }
  public int SourceVersion { get; }
  public int TargetVersion { get; i}
  public int TargetVersionUsedForTmUpdate { get; }
  public bool Updated { get; }
}
```

You specify the following information:

Information	Description
ItemId	The Sitecore item ID of each updated item to send to the remote TM.
SourceVersion	The source-language version of the updated items to send to the remote TM.
TargetVersion	The target-language version of the updated items to send to the remote TM.
TargetVersionUsedForTmUpdate	The version of the updated target items to send to the remote TM.

5. Inspect the `TmUpdateItem.Updated` property to verify that an item has actually been sent out to update a remote TM.

6. If an item was not actually sent out to update a remote TM, investigate the following possible reasons:

   - The target version was not updated after the translation was received.
   - The Connector did not find a translation record associated for the Sitecore item ID and the specified source and target version combination.

7. If neither of the reasons in the previous step is relevant, then check the Clay Tablet and Sitecore log files to determine whether any unexpected errors occurred during the TM update process.

8. To verify which target version was sent to update the remote TM, inspect the `TmUpdateItem.TargetVersionUsedForTmUpdate` property. Typically the Connector uses the target version specified in the arguments to update the remote TM.

   However, if the **Always Update TM using latest target version** check box is selected in the `/sitecore/system/Settings/Clay Tablet Settings/Clay Tablet Settings` item, the pipeline uses the latest target version instead of the specified target version, and `TmUpdateItem.TargetVersionUsedForTmUpdate` returns that version number. If the Connector does not send out item to update a remote TM, then `TmUpdateItem.TargetVersionUsedForTmUpdate` is not specified.

Note: When you use the pipeline to send out multiple items to update a remote TM, then the Connector groups the items in the same way that it grouped the original items for translation in an XML file. Items that were originally in the same XML file (asset task file) are sent out in the same TM update file.

Sample code for calling the pipeline:

```
SendTmUpdatePipelineArgs pArgs = new SendTmUpdatePipelineArgs();
pArgs.SourceLanguage = _itemSourceLanguageName;
pArgs.TargetLanguage = _itemTargetLanguageName;
pArgs.AddTmUpdateItem(_itemID, _itemSourceVersion, _itemTargetVersion);
CorePipeline.Run("SendTmUpdatePipeline", pArgs);
int tmUpdatedCount = 0;
foreach (TmUpdateItem updateItem in pArgs.TmUpdateItems)
{
  if (updateItem.Updated)
     {
  tmUpdatedCount++;
  }
}
```

9.3 Calling Custom Workflow Logic

The Connector can call custom logic when it sends out items for translation and when content returns from translation.

To configure custom logic for when the Connector sends out items for translation:

1. In the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file, add the `<CustomSendItemsPipeline> (after existing <SendItemPipeline>` section to call custom logic after the Connector sends out items for translation. This section can be similar to the following:

```
<CustomSendItemsPipeline>
<processor type="Your.Class, Your.Assembly" method="YourMethod" />
</CustomSendItemsPipeline>
```

2. Ensure that the processor method has the following signature:

```
public void Your.Class.YourMethod (Sitecore.Pipelines.PipelineArgs arg);
```

3. You can retrieve the following information from the `arg` parameter:

Information	Description
arg.CustomData ["Database"] (String)	The Sitecore database that contains the item sent out for translation.
arg.CustomData ["ItemId"] (String)	The identifier of the Sitecore item sent out for translation.
arg.CustomData ["SourceLanguage"] (String)	The Sitecore language code of the source language.
arg.CustomData ["SourceVersion"] (int)	The version of Sitecore item sent out for translation.
arg.CustomData ["TargetLanguage"] (String)	The Sitecore language code of the target language.
arg.CustomData ["TargetVersion"] (int)	The target version of the Sitecore item.

Information	Description
arg.CustomData ["JobId"] (String)	The identifier of the Clay Tablet job that contains the item sent out for translation.
arg.CustomData ["AssetId"] (String)	The identifier of Clay Tablet asset that contains the item sent out for translation. An asset can contain multiple Sitecore items, because the Connector aggregates multiple items into one XML to send out for translation.

To configure custom logic for when the Connector receives translated content:

1. In the `Website_root/Website/App_Config/Include/CT3Translation.config` configuration file, add the `<CustomHandleTranslatedItemPipeline> (after existing <SendItemPipeline>` section to call custom logic after the Connector receives the translated content. This section can be similar to the following:

```
<CustomHandleTranslatedItemPipeline>
<processor type="Your.Class2, Your.Assembly" method="YourMethod2" />
</CustomHandleTranslatedItemPipeline>
```

2. Ensure that the processor method has the following signature:

```
public void Your.Class2.YourMethod2 (Sitecore.Pipelines.PipelineArgs arg);
```

3. You can retrieve the following information from the `arg` parameter:

Information	Description
arg.CustomData ["Database"] (String)	The Sitecore database that contains the item sent out for translation.
arg.CustomData ["ItemId"] (String)	The identifier of the Sitecore item sent out for translation.
arg.CustomData ["TargetLanguage"] (String)	The Sitecore language code of the target language.
arg.CustomData ["TargetVersion"] (int)	The target version of the Sitecore item.
arg.CustomData ["JobId"] (String)	The identifier of the Clay Tablet job that contains the item sent out for translation.

Information	Description
arg.CustomData ["AssetId"] (String)	The identifier of Clay Tablet asset that contains the item sent out for translation. An asset can contain multiple Sitecore items, because the Connector aggregates multiple items into one XML to send out for translation.
arg.CustomData ["AssetTaskId"] (String)	The identifier of Clay Tablet asset task of the received translation. There is a one-to-many relationship between assets and asset tasks, because one source can be translated into multiple target languages.

9.4 Using Custom Logic to Overwrite the Workflow Target-Language Setting when Automatically Sending Items to the Translation Queue via Workflow

The `CustomQueueItemTargetLanguagesPipeline` setting in `CT3Translation.config` enables you to use your own logic to override the target-language assignment when the Connector automatically sends a source to the Translation Queue via workflow.

The following is a sample implementation of the pipeline:

```
public class TestCustomQueueItemTargetLanguagesPipeline
{
  public void Process(QueueItemsTargetLanguagesPipelineArgs args)
  {
    Item item = args.SourceItem;
    if (item.Language.ToString().StartsWith("en"))
    {
      args.TargetLanguages.Add("fr-FR");
      args.TargetLanguages.Add("es-ES");
    }
    else
    {
      args.TargetLanguages.Add("en");
    }
  }
}
```

The previous code sample checks the source item to send to the Translation Queue:

- If the source language starts with `en`, then it adds the item to the queue with `fr-FR` and `es-ES` as target languages.
- Otherwise, it adds the item to the queue with `en` as the target language.

Any implementation should have a method with the signature like:

```
public void Process(QueueItemsTargetLanguagesPipelineArgs args)
```

The parameter is of `ClayTablet.SC.Pipelines.QueueItemsTargetLanguagesPipelineArgs` type:

```
public class QueueItemsTargetLanguagesPipelineArgs : PipelineArgs
{
  public Item SourceItem { get; }
  public List<String> TargetLanguages { get; }
}
```

The implementation can get the `SourceItem` from the argument, and it can insert target language codes into the `TargetLanguages` list of the `QueueItemsTargetLanguagesPipelineArgs` object.

9.5 Using Custom Logic to Configure how the UploadService Creates Jobs when Automatically Sending Out Items from the Translation Queue

The `CustomAutoSendQueuesPipeline` setting in `CT3Translation.config` enables you to use your own logic to determine how to automatically create jobs from the Translation Queue. It works in conjunction with the `AutoSendQueueItemsTime` option in the `ClayTablet.CT3Agent.UploadService` agent, which is described in "Configuring Global Service Settings" on page 39.

The following is a sample implementation of the pipeline:

```
public class TestCustomAutoSendQueuesPipeline
{
  public void Process(AutoSendQueueJobsArgs args)
  {
    string frJobId = null;
    string otherJobId = null;

    foreach (TranslationQueueInfo queueItem in args.TranslationQueueItems)
    {
      TranslationJobInfo jobInfo;
      if (queueItem.ItemTargets.Contains("fr-FR"))
      {
        if (frJobId == null)
        {
          jobInfo = args.AddNewJob();
          // Create a separate job for queue items with fr-FR as target languages
          jobInfo.Name = "Auto-sent French job " + DateTime.Now.ToString
          ("MM/dd/yyyy HH:mm");
          jobInfo.Description = "Automatically send fr-FR in a separate job
          scheduled at " + args.ScheduledSendTime.ToString("MM/dd/yyyy HH:mm");
          jobInfo.DueDate = DateTime.Now.AddMonths(1);

          // You can also specify a certain LSP account to send the job to
          // jobInfo.SendAccount = "FR LSP";
          frJobId = jobInfo.ID;
        }
        else
        {
          jobInfo = args.GetJob(frJobId);
        }
      }
      else
      {
        if (otherJobId == null)
        {
          // Create a job for all other queued items
          jobInfo = args.AddNewJob();
          // Not specify job meta information, the connector will automatically
          fill in the information based on AutoSendQueueItems* configurations
          // in CT3Translation.config
          otherJobId = jobInfo.ID;
        }
        else
        {
          jobInfo = args.GetJob(otherJobId);
        }
      }
      // If the queue item's ID is not inserted in a TranslationJobInfo object,
      the queue item will be skipped and left in the queue
      jobInfo.QueueIdsInTheJob.Add(queueItem.ID);
    }
  }
}
```

The previous code sample checks the queued items and creates a translation job named `Auto-sent French job` with a timestamp. This job includes all queued items with `fr-Fr` as one of its target languages. The Connector's default job contains all other queued items.

This enables you to submit different queued items in different jobs, and you can send each job using a different producer key. This also supports sending different queue items at different times, by leaving out some of the queued items in one invocation of the pipeline and picking them up in a later invocation, using the `args,ScheduledSendTime` to determine which ones to send or leave out.

Any implementation should have a method with the signature like:

```
public void Process(AutoSendQueueJobsArgs args)
```

The parameter is of `ClayTablet.SC.Pipelines.AutoSendQueueJobsArgs` type:

```
public class AutoSendQueueJobsArgs : PipelineArgs
{
  public List<TranslationQueueInfo> TranslationQueueItems;
  public DateTime ScheduledSendTime;
  public List<TranslationJobInfo> TranslationJobsToCreate;
  public TranslationJobInfo AddNewJob();
  public TranslationJobInfo GetJob(string id);
}
```

Implementation can iterate over `TranslationQueueItems` to check all the queued items waiting to be sent out. The type of `ClayTablet.SC.Pipelines.TranslationQueueInfo` provides information about the queued item:

```
public class TranslationQueueInfo
{
  public String ID { get; }
  public String ItemID { get; }
  public String ItemDatabase { get; }
  public String ItemSource { get; }
  public int ItemSourceVersion { get; }
  public String ItemPath { get; }
  public List<String> ItemTargets { get; }
  public DateTime CreateTime { get; }
  public bool SyncTranslation { get; }
  public String SitecoreUser { get; }
}
```

The implementation should either:

- Insert each queued item into a `ClayTablet.SC.Pipelines.TranslationJobInfo` object, which will cause it to be sent out as part of the job.
- Leave a queued item alone, so it will remain in the Translation Queue for this scheduled automatic sending invocation.

```
public class TranslationJobInfo
{
  public string ID { get; }
  public string Name { get; set; }
  public string Description { get; set; }
  public string SendAccount { get; set; }
  public DateTime? DueDate { get; set; }
  public string SendUser { get; set; }
  public List<String> QueueIdsInTheJob { get; }
}
```

A new `TranslationJobInfo` is created by `AutoSendQueueJobsArgs.AddNewJob()`. The implementation can set the job's metadata, including name, description, sending account (producer key), sending user (the Sitecore user in the `Creator` metadata of the job), and due date. Insert the queue item's ID into `TranslationJobInfo.QueueIdsInTheJob`. After creating a `TranslationJobInfo`, the implementation should typically save it or its ID for reuse when iterating over other queued items that should be added into the same job. The `AutoSendQueueJobsArgs.GetJob` (string) enables the implementation to retrieve a previously created `TranslationJobInfo` so that it can add more queued items into the same job.

9.6 Using Custom Logic to Populate the Translation Queue

You can now call the `<SendToTranslationQueuePipeline>` pipeline to put Sitecore items into the Translation Queue.

To call the pipeline:

You create an instance of the `ClayTablet.SC.Pipelines.SendToTranslationQueuePipelineArgs` class to pass to the pipeline as argument.

The `SendToTranslationQueuePipelineArgs` class is defined in `ClayTablet.SC.dll` as:

```
public class SendToTranslationQueuePipelineArgs : PipelineArgs
{
  public String SourceLanguage { get; set; }
  public String[] TargetLanguages { get; set; }
  public List<String> ItemIds { get; }
  public int ItemsQueued { get; }
}
```

This class enables you to specify the source language and target languages for the items, and add Sitecore item ID into the `ItemIds` list. The Connector uses the latest source version of the items. When the Connector sends out content for translation from Translation Queue, it creates new target versions for these items. After calling the pipeline, you can use the `ItemsQueued` property to check how many Sitecore items are actually added to the Translation Queue.

The following is sample code for calling the pipeline:

```
static public int SendToTranslationQueue(string sourceLang, string[] targetLangs,
string[] itemIds)
{
  SendToTranslationQueuePipelineArgs args = new SendToTranslationQueuePipelineArgs
  ();
  args.SourceLanguage = sourceLang;
  args.TargetLanguages = targetLangs;
  args.ItemIds.AddRange(itemIds);
  CorePipeline.Run("SendToTranslationQueuePipeline", args);
  return args.ItemsQueued;
}
```

Index