Clay Tablet Translation

SDK for .NET Programming Guide

Version 3.5.5

May 18, 2018

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 Technologies SDK for .NET

The Clay Tablet Technologies Software Development Kit (SDK) for .NET describes the .NET technologies that Clay Tablet Technologies provides for developers to create a Connector to the Clay Tablet Translation Platform.

Note: The Clay Tablet Technologies Software Development Kit (SDK) is also available for Java. For details, refer to the *Clay Tablet Translation SDK for Java Programming Guide*.

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, a Lionbridge company, the corporate entity that publishes the Connector and the Clay Tablet Platform.
Clay Tablet Platform	The hosted (IaaS) connectivity platform that receives and routes content from content management systems (CMSs) to translation providers and back during implementation.
Connectors	Discrete binaries that are installed on the CMS or TMS side to provide connectivity to the CTT Platform. This SDK helps you code a Connector.
IaaS	Infrastructure as a Service. The Clay Tablet Platform is an IaaS, because it is a hosted platform.
On-Premise Platform	A version of the Clay Tablet Platform that is hosted on the premises of the Clay Tablet client, instead of on AWS.
Producer	CMS, database, or another system that sends content or documents out for translation.
Provider	Translation system or technology that provides translation or translation management 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.
WSE 2	Web Services Enhancements (WSE) 2.0 from Microsoft. WSE2 must be installed on the machine where Producer or Provider code will run. For more information about WSE2, see: http://www.microsoft.com/downloads/details.aspx?familyid=fc5f06c5-821f-41d3-a4fe-6c7b56423841&displaylang=en

1.2 What is the Clay Tablet Translation Solution?

Clay Tablet is the easiest, most flexible way to integrate content systems with translation providers and their translation software systems. Clay Tablet's open, flexible, scalable, hub-style architecture supports the integration of any content system with any translation system or technology.

In Clay Tablet:

- The hub of this solution is the *Clay Tablet Translation Platform*.
- The spokes of this solution are the *Connectors*. A Connector connects either between a content management system and the Platform, or between a translation provider and the Platform.

1.2.1 What Is the Clay Tablet Translation Platform?

Clay Tablet Platform is the hosted (IaaS) connectivity platform that receives and routes content from content systems to translation providers and back. It is hosted on Amazon Web Services (AWS). 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.

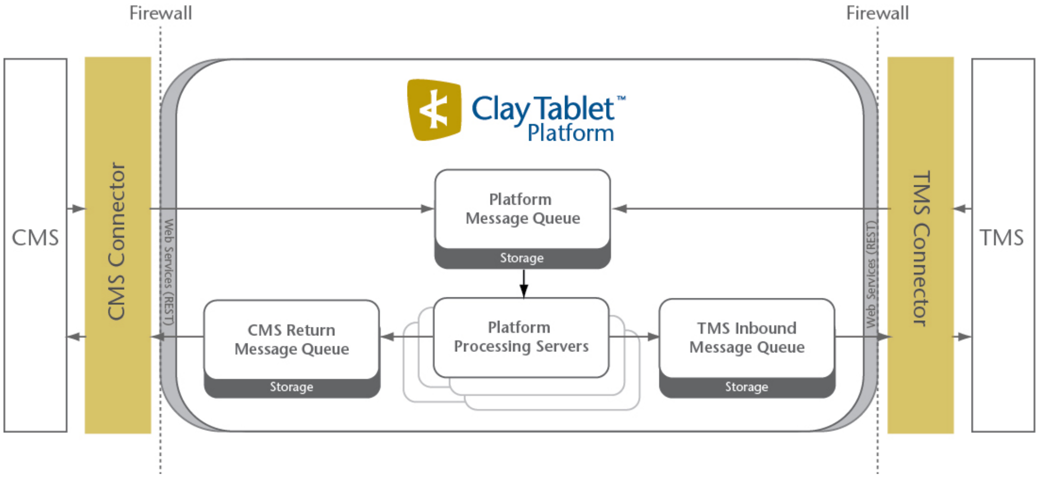
1.2.2 What is a Connector?

A Connector is an application with an interface to the Clay Tablet Translation Platform. Depending on the type of Connector you build, it connects to either a content system, such as a CMS, or a translation provider or translation management system (TMS):

- If you build a Connector between a *content system* and the Platform, this is a *Content* Connector, because the content system produces the content for translation.
- If you build a Connector between a *TMS* and the Platform, this is a *Translation* Connector, because the TMS provides the translated content.

You use this SDK to build a Connector between your content system or translation management system and the Clay Tablet Translation Platform.

1.2.3 How Does Clay Tablet Work?



Clay Tablet works with existing infrastructure, technology, and code bases, which means faster implementations and preserving existing technology investments.

Note: This SDK contains information for coding connections to *both* Content Systems (content producers) *and* translation systems (translation providers). You probably will only be actually coding an interface for one side (content *or* translation), but this guide provides the information for both – so you can understand the functions of the other side as well.

You can ensure your Connector works with the latest Amazon AWS services using the libraries provided by Clay Tablet as part of this SDK package.

1.3 Using this Guide

Purpose of this guide

This guide describes everything you need to know to build a Connector to the Clay Tablet Platform, either from a CMS or a TMS.

It contains overview information and step-by-step, how-to procedures for creating a Connector to the Clay Tablet Translation Platform. It includes several code examples as well as some tips and best practices for using this development platform.

Who should use this guide

This guide is intended for developers at a CMS, TMS, or translation provider who want to build a Connector to the Clay Tablet Platform using .NET.

Both content producers and translation providers use the SDK to build Connectors to the Clay Tablet Translation Platform.

How content producers use the SDK

Content producers use the SDK to build a *Content Connector*. This Connector sits between the Content Producer application, such as a CMS, and the Clay Tablet Translation Platform. It has the following functionality:

- sending content for translation
- sending metadata about translation jobs
- grouping assets into a group, in order to send multiple content items for translation as a single job
- retrieving updates about translation status
- retrieving translated content

How translation providers use the SDK

Translation providers use the SDK to build a *Translation Connector*. This Connector sits between the Translation Producer application, such as a TMS, and the Clay Tablet Translation Platform. It has the following functionality:

- receiving content for translation
- receiving metadata about translation jobs
- sending translation status updates
- sending translated content back to the content producer

What you should already know

This document assumes that:

- You are familiar with .NET programming.
- Your company is a CMS or TMS publisher, or a translation provider.

How this guide is organized

This guide contains the following chapters:

Chapter	Description
"Welcome to the Clay Tablet Technologies SDK for .NET" on page 7	A brief description of the Clay Tablet Translation solution and how it works, including an explanation of its components: the Clay Tablet Platform and Clay Tablet Translation Connectors. It also includes information about this guide and Clay Tablet Technologies Support contact information.

Chapter	Description
"Event Overview" on page 25	The Connector you build uses events, which are .NET classes, to send content and other data to the Clay Tablet Platform and to retrieve content and other data from it. This chapter provides an overview of the events in the context of the objects that they use.
"Events for Building a Content Connector" on page 40	The classes (events) in this section are relevant when a Content Producer builds a Connector.
"Events for Building a Translation Connector" on page 92	The classes (events) in this section are relevant when a Translation Provider builds a Connector.
"Events that All Connectors Receive" on page 151	The classes (events) in this section are relevant when either a Content Producer or a Translation Provider builds a Connector.
"Job-Related Interfaces and Metadata-Related Classes" on page 155	These interfaces group content into jobs and provide job-related metadata.
"Testing Your Connector" on page 166	How to test the Connector you build.
"Appendix: High-Level .NET Reference Information" on page 167	Reference information about the abstract classes, the Language and Account classes, interfaces, and return types used by the events described in this SDK.

How to find out more about Clay Tablet Translation

Sample code is included in the `samples` folder in the SDK package.

An SDK and guide are also available for Java.

Documentation conventions

This guide uses the following conventions:

Convention	Description	Example
Bold	Names of objects, classes, methods, properties, events, and settings.	To handle a **StartAssetTask** event ...
`Courier`	Sample code.	`AcceptAssetTask acceptAssetTask_event = new AcceptAssetTask();`
Italics	Highlights terms to emphasize, variables, or document titles.	Rejects the translation to a single target language (an *asset task*).

Additional notes

Sample code in this guide is provided in C#.

The coding style in this SDK may seem non-standard in some areas (e.g. using `get()`/`set()` methods instead of the .NET property, `get`/`set`). This is done to simplify keeping our Java and .NET SDKs synchronized. We apologize for any unnecessary confusion.

1.4 Licensing Information

The contents of this SDK and all related materials, are licensed intellectual property of Clay Tablet Technologies. If you are unsure of the terms of usage related to these materials, please contact your Clay Tablet Account Manager.

To obtain a license, please contact Clay Tablet Sales:

Email @: sales@clay-tablet.com

Telephone: +1.416.363.0888 option "2"

1.5 How to Contact Clay Tablet Support

Support is available to licensed users only under the terms of their support contract. If you are in the process of evaluating this SDK for purchase and require assistance, please contact your Clay Tablet Account Manager.

To contact Clay Tablet Development Support:

Email @: dev.support@clay-tablet.com

1.6 Feedback and Comments

We are always interested in receiving feedback and suggestions regarding our products and supporting materials, such as this SDK. Please send us your comments or suggestions by e-mail to support@clay-tablet.com.

2 Connector Core Features

The information in this section is based on the standards that Clay Tablet created while developing its own Connectors. They type of Connector you build determines which core features you should incorporate:

- "Content Connectors–Core Features"
- "Translation Connectors–Core Features"

2.1 Content Connectors–Core Features

Version 1.0 Features

To be production ready, your first version of the Connector requires certain features:

Connector installer

Typically, the content system defines how to install the Connector. Clay Tablet uses the following standards while respecting requirements of the content system:

Priority	Feature
Requirement	Install, uninstall, and upgrade the Connector.
Recommendation	Enable and disable the Connector. Most content systems support active and inactive software components.
Nice to Have	The installer does not require restarting the content system.

Licensing-server support

Priority	Features
Requirement	The Connector must interact with and obtain license keys from the Clay Tablet License Server.

Database support

The Connector must have a database (DB) installed, create associated tables, and enable communication between the Connector and the DB.

Clay Tablet typically provides support for only one DB vendor as part of an initial Connector software release, and it collaboratively determines the most appropriate Relational Database Management System (RDBMS) on a case-by-case basis, very early in the development cycle.

Priority	Features
Requirement	Installation of a Clay Tablet-supported RDBMS. **Note:** The DB software installation can be separate from the content system installer. However, the Connector must provide DB scripts.
Requirement	The Connector needs a configuration interface or files to communicate and interact with the RDBMS to enable DB setup and use.
Requirement	The RDBMS is upgradeable.
Requirement	There must be a mechanism for recording and annotating application database version information. The Connector must remain in sync with its DB schema, which means the application must track and update its version information to facilitate upgrades.

Temporary file storage facility

Historically, Connectors temporarily stored files on the local file system while processing jobs. For various technical and implementation reasons, this is no longer permitted.

Priority	Features
Requirement	The Connector must store all temporary files and information in the Clay Tablet database. Using the local file system for temporary file storage is not permitted.

Logging

The Connector logs transaction information, including errors.

Priority	Features
Requirement	The Connector must write transaction information to a log file.
Requirement	The Connector must create a new log file each day.
Requirement	The Connector must support multiple logging levels, because Clay Tablet supports two logging levels: ■ `Default` (Informational, Warnings, Errors) ■ `Debug` (verbose logging for all Clay Tablet activities)

Connector configuration options

The Connector has configuration options that an administrator can modify to affect the software performance.

Priority	Features
Requirement	An administrator can specify configuration options via an editable configuration file stored locally on the file system to control the application settings listed below. For the initial release, an administration graphical user interface (GUI) is not required.
Requirement	(Configuration option) The Connector supports defining the maximum number of Items in an XML file: The Connector must support up to 100 items in a Clay Tablet XML file (translation job).
Requirement	(Configuration option) An administrator can set the application logging level to either `Default` or `Debug`. The default level is Default. For details, see "Logging", above.
Requirement	(Configuration options) An administrator can set the following standard proxy implementation settings: ■ proxy server address ■ port ■ protocol
Nice To Have	An administrator can change configuration options without restarting the content system.

Language mapping

Priority	Features
Requirement	The Connector must use a dedicated configuration file to map content-system languages to Clay Tablet language codes.

Application versioning

Keeping the application and database versions synchronized is a critical component of normal Connector operations. It also facilitates support and troubleshooting.

Priority	Features
Requirement	The Connector must write its version information to the log files on instantiation.
Requirement	The Connector must write the version information for its database implementation to the log files on instantiation.

User interfaces

Some user interfaces (UIs) are mandatory in the Connector. The UIs should integrate with the design and operational considerations of the content system.

Priority	Features
Requirement	The user can select a single item and send it directly to translation or to the Translation Queue.
Requirement	There is a Translation Queue, where a user must approve an item to send it out for translation.
Requirement	There is Bulk Translation feature, where the user can select and send multiple items for translation in a few steps.
Requirement	There is a Translation Status UI, where the user can monitor the translation status of items sent for translation.

Basic functionality

The Connector must include the following functionality across the UIs described above. Each content system may implement these features differently.

Priority	Features
Requirement	The user can specify the source language for a translation job.
Requirement	The user can specify the target language for a translation job.
Requirement	The user can specify multiple target languages for a translation job.
Requirement	The user can search for content items by string.
Requirement	The user can search for content Items, including child items, by content tree.
Requirement	The user can select one, multiple, or all items from a result set to send for translation.
Requirement	The user can send items to Translation Queue.
Requirement	The user can send items immediately for translation.
Requirement	The user can define job metadata: The standard base set includes: job name, job description, purchase order, and producer ID.
Requirement	The user can check job status via the Translation Status UI.

Notifications

The Connector uses email notifications to send job information to administrators and end users.

Priority	Features
Requirement	The Connector sends notifications when it sends out a job for translation.
Requirement	The Connector sends notifications when a job returns from translation.
Requirement	An administrator can define email-server information, including hostname, credentials and protocols.
Requirement	The Connector notification system must support using secure SMTP implementations.

Post 1.0 Features

The following features are typically added to subsequent releases. If resource or timelines allow, or client requirements are sufficiently intense, consider including the following functionality:

Select translation provider

This important feature enables end-users to select the translation provider for a job.

Recommendation: Add this feature as soon as possible after the 1.0 release.

Graphical administration interface

Early-version Connectors can be driven primarily by values in configuration files. Where technically possible, consolidate these values in one file.

Graphical language-mapping interface

Mapping languages is a common task.

Recommendation: Create a GUI representing the language-mapping logic in the configuration files of the Connector.

Additional database considerations

Provide additional upgrade support by creating upgrade packages that query existing Connector installations for version information, and apply appropriate upgrade logic.

Filters

Filtering helps users control cost, by ensuring only appropriate content is sent for translation.

Recommendation: Enable users to filter result sets by criteria, including strings and dates.

Metadata

Consider expanding the default metadata set, by adding support for defining custom metadata elements.

Logging interface

Provide a UI to find and consume, and possibly download log files.

Notification capabilities

Extend the default feature set. Additional features can include notifications when jobs fail, etc.

Team profiles

Enable administrators to create teams, which enable separating groups of users and their content, jobs, and status.

Translation-delivery options

Enable end-users to send a job to either generate a quote from the translation provider or to calculate the word count.

Enhanced translation status functionality and UI

- Enable end-users to delete translation jobs (both complete and incomplete).
- Enable end-users to hide completed jobs from the UI after a configurable time period (for example, X days).

Update a remote translation-memory server

Enable users to deliver TMX files, possibly selected in a UI, to update a remote TM.

Local TM implementation

This feature varies depending on the content system. The goal is to store translated content locally, then compare translation requests, or compare field-level time stamps, if that functionality exists.

2.2 Translation Connectors–Core Features

Version 1.0 Features

To be production ready, your first version of the Connector requires certain features:

Connector installer

Typically, the translation system defines how to install the Connector. Clay Tablet uses the following standards while respecting requirements of the translation system:

Priority	Feature
Requirement	Install, uninstall, and upgrade the Connector.

Priority	Feature
Recommendation	The Connector can run as a service on the operating system.
Nice to Have	The installer does not require restarting the translation system.

Temporary file storage facility

The Connector temporarily stores files on the local file system, where they run.

Priority	Features
Optional	Temporarily storing files on the local file system facilitates debugging.
Recommendation	The Connector can remove a file when it receive an `ApprovedAssetTask` message for the corresponding asset, indicating that the

The Clay tablet connectors will do so indefinitely or until the provider receives an ApprovedAssetTask message for that asset, then it will remove it.

Logging

The Connector logs transaction information, including errors.

Priority	Features
Requirement	The Connector must write transaction information to a log file.
Requirement	The Connector must create a new log file each day.
Requirement	The Connector must support multiple logging levels, because Clay Tablet supports two logging levels: - `Default` (Informational, Warnings, Errors) - `Debug` (verbose logging for all Clay Tablet activities)

Connector configuration options

The Connector has configuration options that an administrator can modify to affect the software performance.

Priority	Features
Requirement	An administrator can specify configuration options via an editable configuration file stored locally on the file system to control the application settings listed below.
Requirement	(Configuration option) An administrator can set the application logging level to either `Default` or `Debug`. The default level is Default. For details, see "Logging", above.

Priority	Features
Nice To Have	(Configuration options) An administrator can set the following standard proxy implementation settings: - proxy server address - port - protocol

Language mapping

Priority	Features
Requirement	The Connector must use a dedicated configuration file to map content-system languages to Clay Tablet language codes.

Application versioning

Having an application version is a critical component of normal Connector operations. It also facilitates support and troubleshooting.

Priority	Features
Requirement	The Connector must write its version information to the log files on instantiation.

Basic functionality

The Connector must include the following functionality:

Priority	Features
Requirement	The Connector must process and respond to the StartAssetTask message. **Note:** The Connector must delete all messages after processing them, so that they do not remain in the queue indefinitely.
Requirement	The Connector must receive and process the following messages: - ApprovedAssetTask - AcknowledgedAssetTaskTranslation - RejectedAssetTask - RejectedAssetTaskQuote - StartAssetTask * Required for minimum viable product **Note:** The Connector must delete all messages after processing them, so that they do not remain in the queue indefinitely.

Priority	Features
Requirement	The Connector must send the following messages: - AcceptAssetTask * Required for minimum viable product - AssetTaskProviderError - SubmitAssetTask * Required for minimum viable product - UpdateAssetTaskState

Notifications

The Connector uses email notifications to send job information to administrators and end users.

Priority	Features
Nice to have	The Connector sends notifications to a specified list when it sends out a job for translation.
Nice to have	The Connector sends notifications to a specified list when a job returns from translation.
Nice to have	The Connector sends notifications to a specified list when there is an error, such as a failure to add a job into a TMS due to an invalid language code.
Nice to have	An administrator can define email-server information, including hostname, credentials and protocols.
Nice to have	The Connector notification system must support using secure SMTP implementations.

Post 1.0 Features

The following features are typically added to subsequent releases. If resource or timelines allow, or client requirements are sufficiently intense, consider including the following functionality:

Enhanced functionality

Consider adding the following functionality:

Priority	Features
Requirement	The Connector must process the following messages: **Note:** The Connector must delete all messages after processing them, so that they do not remain in the queue indefinitely. - ApprovedAssetTaskQuote - ApprovedTranslationCorrectionAssetTask - CanceledAssetTask - CanceledSupportAsset - StartAssetTaskQuote - StartNeedAssetTaskUpdate - StartNeedTranslationCorrectionAssetTask - StartSupportAsset - StartUpdateTMAsset
Requirement	The Connector must send the following messages: - AcceptNeedTranslationCorrectionAssetTask - AcceptUpdateTMAsset - SubmitTranslationCorrectionAssetTask - SubmitUpdatedAssetTask

Metadata

Consider expanding the default metadata set, by adding support for translation-provider specific custom metadata elements, such as analysis codes.

Notification capabilities

Extend the default feature set. Additional features can include notifications when jobs fail, etc.

Update a remote translation-memory server

Enable users to deliver TMX files, possibly selected in a UI, to update a remote TM.

2.3 Sending a Heartbeat Message to the Clay Tablet Platform

You can periodically send heartbeat messages to the Clay Tablet Platform so that it can identify the type and location of your connector. The heartbeat messages can also help the Clay Tablet Platform to detect common connector problems, such as the same key being used by multiple instances of connectors, or connectors that are unavailable. The following code sample illustrates how to send a heartbeat message to the Clay Tablet Platform:

```
using com.claytablet.model.Event;
using com.claytablet.provider;
```

```
SourceAccountProvider sap = new SourceAccountProvider(sourceAccountXML);
TargetAccountProvider tap = new TargetAccountProvider(targetAccountXML);
HeartBeat heartbeat = new HeartBeat();
heartbeat.setSignature("Name of your connector");
// sessionStartTime tracks the time when this instance of connector is first
started
heartbeat.setSessionStartTime(sessionStartTime);
// HealthState is an enum reporting the state of the connector
// use HealthState.Normal to report normal state
// there are also Unknown, Warn, Error, Fatal states you can report
heartbeat.setHealthState(HealthState.Normal);
heartbeat.setHostName("Your host name");
heartbeat.setIP("Your host ip");
heartbeat.setSourceAccountId(sap.get().getId());
heartbeat.setTargetAccountId(tap.get().getId());
sender.sendEvent(heartbeat);
```

3 Event Overview

The Connector you build uses events, which are .NET classes, to send content and other data to the Clay Tablet Platform and to retrieve content and other data from it. Each event controls one of the objects described below. Although the events that control these objects are not related in the .NET hierarchy, looking at the events in the context of these objects is helpful in understanding how the entire system functions.

Object	**Description**	**For Information about Events Using These Objects, See...**
Asset	A content item for translation.	page 26
SupportAsset	A supporting document and its metadata. Support assets are not translated by the translation provider, but they provide helpful context for the translator.	page 27
AssetTask	Asset tasks are either assets (source content for translation) or target-specific translation tasks for translating a particular content item (asset). The task of translating a particular asset (content item) into a single language is a single asset task. Therefore suppose a content item will be translated into both French and German. The asset is the content item in the source language. There are three asset tasks: the original content item; the translation of the content item into French; and the translation of the content item into German.	page 28
AssetTaskQuote	A request for a quote when the Content Connector sends content for translation, or the quote itself.	page 31
AssetTaskState	The state of an asset task (translation task).	page 33
AssetTaskTranslation	An asset task that has been translated (translated content).	page 34
Project	A group of assets with the same Project ID. **Note:** You can also group assets as a job. For a comparison of jobs and projects, see "Grouping Assets for Translation" on page 40.	page 34
AssetTaskUpdate	The request for an update to previously translated content.	page 35
UpdatedAssetTask	Updated translated content.	page 36

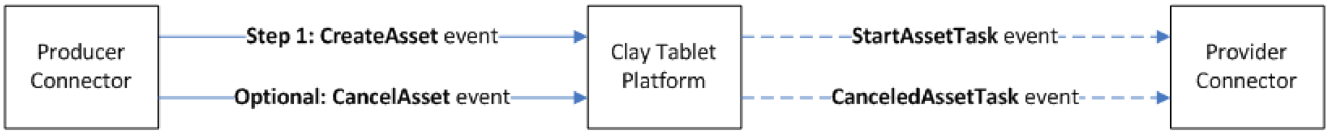
Object	Description	For Information about Events Using These Objects, See...
Translation CorrectionAsset	The request to correct a translation, and later it is the actual corrected translation.	page 37
UpdateTMAsset	Translated content that the content producer sends to the translation provider so that it can update the translation memory (TM).	page 38
ProcessingError	Processing errors can occur when the Clay Tablet Platform is processing an event it receives.	page 38

3.1 Asset Events

The Content Connector creates *assets* (`Assets`), which are content or documents that it sends for translation. The Content Connector sends events about these assets to the Clay Tablet Platform, which then creates *asset tasks* (`AssetTasks`) to send to the Translation Connector. There is a separate asset task for each asset and target combination. For example, if an asset is being translated into two languages, it has two corresponding asset tasks.

For more information on `AssetTask` events, see page 28.

The following diagram shows how these events work together in sequence:



Note: In the diagram above, events with dotted lines are not asset events, so they are not described in this section.

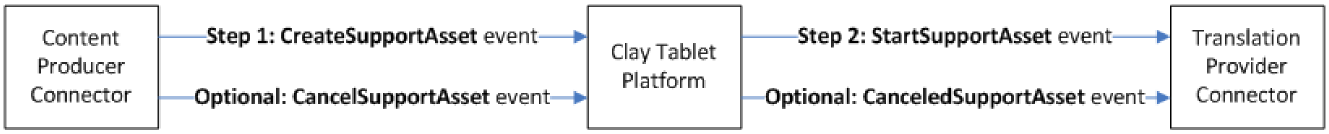
Event	Content Connector	Clay Tablet Platform	Description
CreateAsset	sends	receives	The Content Connector creates an asset to send for translation. This asset is content or a file.

Event	Content Connector	Clay Tablet Platform	Description
CancelAsset	sends	receives	The Content Connector cancels an asset that it previously sent for translation. The Platform then launches events to notify the provider of the cancellation (CanceledAssetTask) and remove any files that have been uploaded for the asset.

3.2 SupportAsset Events

A support asset (`SupportAsset`) is a document that the content producer sends to the translation provider to provide helpful context for the translator. It is not translated by the translation provider.

The following diagram shows how SupportAsset events work together in sequence:


Content Producer Connector
Step 1: CreateSupportAsset event
Optional: CancelSupportAsset event
Clay Tablet Platform
Step 2: StartSupportAsset event
Optional: CanceledSupportAsset event
Translation Provider Connector

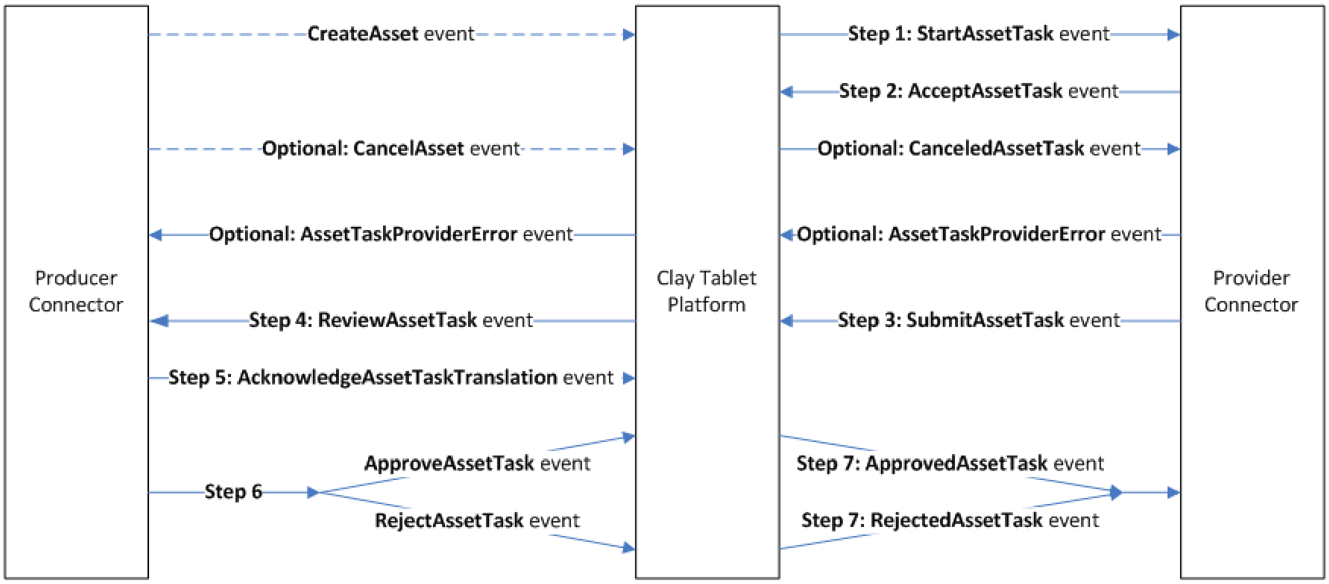
Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
CreateSupportAsset	sends	receives		The Content Connector launches this event to create a new support asset.
StartSupportAsset		creates	retrieves	The Clay Tablet Platform launches this event to send a new support asset to the Translation Connector.
CancelSupportAsset	sends	receives		The Content Connector launches this event to cancel a support asset it previously created.
CanceledSupportAsset		creates	retrieves	The Clay Tablet Platform launches this event to notify the Translation Connector that the support asset is canceled and to remove the file that was uploaded for the support asset.

3.3 AssetTask Events

The Clay Tablet Platform creates an *asset task* from assets it receives from the Content Connector. An asset task represents a work task that the translation provider performs for an asset.

Asset tasks are either assets (source content for translation) or target-specific translation task for translating a particular content item (asset). The task of translating a particular asset (content item) into a single language is a single asset task. Therefore suppose a content item will be translated into both French and German. The asset is the content item in the source language. There are three asset tasks: the original content item; the translation of the content item into French; and the translation of the content item into German.

The following diagram shows how `AssetTask` events work together in sequence:


Producer Connector
Clay Tablet Platform
Provider Connector
CreateAsset event
Optional: CancelAsset event
Optional: AssetTaskProviderError event
Step 4: ReviewAssetTask event
Step 5: AcknowledgeAssetTaskTranslation event
Step 6
ApproveAssetTask event
RejectAssetTask event
Step 1: StartAssetTask event
Step 2: AcceptAssetTask event
Optional: CanceledAssetTask event
Optional: AssetTaskProviderError event
Step 3: SubmitAssetTask event
Step 7: ApprovedAssetTask event
Step 7: RejectedAssetTask event

Note: For a larger version of this image, which is suitable for printing, view it in the Clay Tablet Online Help, at http://www.clay-tablet.com/help/Content/Topic_Files/SDK_Platform_Events/Events_-_AssetTask_NET.htm.

Note: In the diagram above, events with dotted lines are not asset-task events, so they are not described in this section.

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	StartAssetTask		creates	retrieves	The Clay Tablet Platform notifies the Translation Connector of a translation task.

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
2	AcceptAssetTask		receives	sends	The Translation Connector notifies the Clay Tablet Platform that it accepts the translation task.
3	SubmitAssetTask		receives	sends	The Translation Connector notifies the Clay Tablet Platform that the translation provider has completed the translation task, and it sends it for review.
4	ReviewAssetTask	retrieves	creates		The Clay Tablet Platform notifies the Content Connector that the translation task is complete, and it sends it for review.
5	AcknowledgeAssetTaskTranslation	sends	receives		The Content Connector acknowledges that it has successfully received translated content in a single target language (asset task).

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
6	ApproveAssetTask	sends	receives		The Content Connector launches this task to indicate that the content producer has reviewed the translation and approves it.
	RejectAssetTask	sends	receives		The Content Connector launches this task to indicate that the content producer has reviewed the translation and rejects it.
7	ApprovedAssetTask		creates	retrieves	The Clay Tablet Platform indicates to the Translation Connector that the content producer approves the translation.
	RejectedAssetTask		creates	retrieves	The Clay Tablet Platform indicates to the Translation Connector that the content producer rejects the translation.

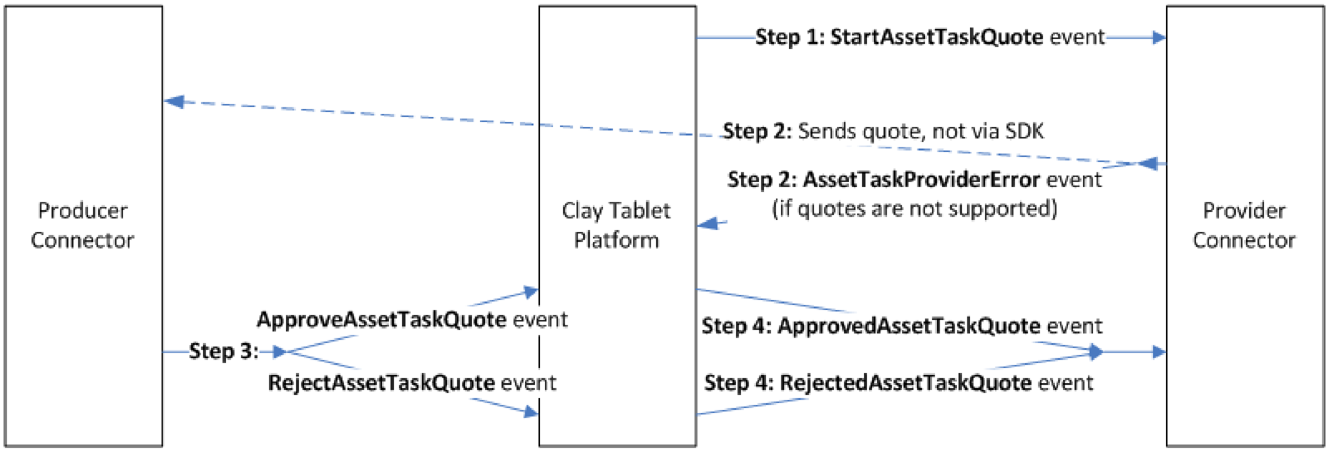
Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
optional	CanceledAssetTask		creates	retrieves	If the Content Connector informs the Platform that it canceled an asset task, the Platform informs the Translation Connector. It also cancels the related asset task (translation).
optional	AssetTaskProviderError	retrieves	■ creates ■ receives	sends	If the Translation Connector encounters a problem during translation (while processing the asset task), it informs the Platform, which then informs the Content Connector.

3.4 AssetTaskQuote Events

The Content Connector can request a quote when it sends content for translation. This is an *asset task quote*. The Clay Tablet Platform sends this request to the Translation Connector.

Note: Use these events only if the translation provider supports providing quotes.

The following diagram shows how AssetTaskQuote events work together in sequence:


Producer Connector
Clay Tablet Platform
Provider Connector
Step 1: StartAssetTaskQuote event
Step 2: Sends quote, not via SDK
Step 2: AssetTaskProviderError event
(if quotes are not supported)
ApproveAssetTaskQuote event
Step 3:
RejectAssetTaskQuote event
Step 4: ApprovedAssetTaskQuote event
Step 4: RejectedAssetTaskQuote event

Note: In the diagram above, the step 2 with a dotted line is not an SDK event.

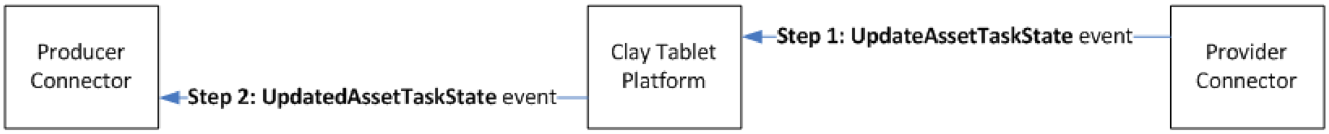
Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	StartAssetTaskQuote		creates	retrieves	The Clay Tablet Platform sends a request for a quote to the Translation Connector, based on a request from the Content Connector.
2					The translation provider sends the quote to the content producer, outside this SDK.
	AssetTaskProviderError		receives	sends	If the translation provider does not provide quotes, it sends this event.
3	ApproveAssetTaskQuote	sends	receives		The Content Connector informs the Clay Tablet Platform that it accepts the quote.
	RejectAssetTaskQuote	sends	receives		The Content Connector informs the Clay Tablet Platform that it rejects the quote.

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
4	ApprovedAssetTaskQuote		creates	retrieves	The Clay Tablet Platform informs the Translation Connector that the content producer accepts the quote. The Platform instructs the Connector to start processing the asset task (start the translation work).
	RejectedAssetTaskQuote		creates	retrieves	The Clay Tablet Platform informs the Translation Connector that the content producer rejects the quote.

3.5 AssetTaskState Events

An AssetTaskState indicates the state of an asset task (translation task).

The following diagram shows how AssetTaskState events work together in sequence:

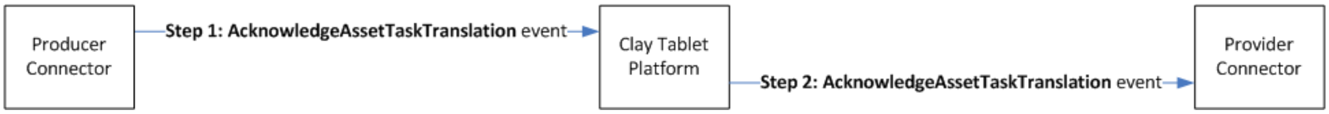

Producer Connector
Step 2: UpdatedAssetTaskState event
Clay Tablet Platform
Step 1: UpdateAssetTaskState event
Provider Connector

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	UpdateAssetTaskState		receives	sends	The Translation Connector sends this event to the Clay Tablet Platform to provide an update about the progress of the asset task (translation).
2	UpdatedAssetTaskState	retrieves	creates		The Clay Tablet Platform sends the update about the asset task (translation) to the Content Connector.

3.6 AssetTaskTranslation Events

An `AssetTaskTranslation` is an asset task that has been translated (translated content).

The following diagram shows how AssetTaskTranslation events work together in sequence:

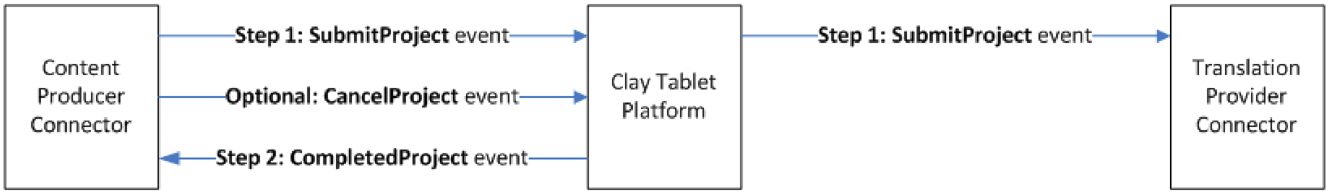

Producer Connector
Step 1: AcknowledgeAssetTaskTranslation event
Clay Tablet Platform
Step 2: AcknowledgeAssetTaskTranslation event
Provider Connector

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	Acknowledge AssetTaskTranslation	sends	receives		The Content Connector sends this event to the Clay Tablet Platform when it acknowledges receiving the translated content. **Note:** This is not the same as the Content Provider accepting the translation, which is the `ApproveAssetTask` event.
2	Acknowledged AssetTaskTranslation		creates	retrieves	The Clay Tablet Platform informs the Translation Connector that the Content Connector acknowledged receiving the translated content.

3.7 Project Events

A *project* is a set of assets that are submitted together for translation.

The following diagram shows how project events work together in sequence:

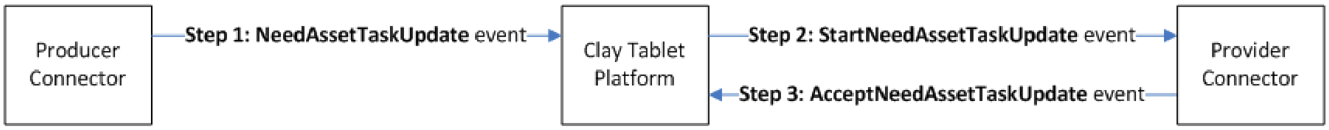

Content Producer Connector
Step 1: SubmitProject event
Optional: CancelProject event
Step 2: CompletedProject event
Clay Tablet Platform
Step 1: SubmitProject event
Translation Provider Connector

Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
SubmitProject	sends	- creates - receives	retrieves	- The Content Connector launches this event when it submits a project for translation. - The Clay Tablet Platform launches this event when it bundles together a set of assets to send for translation.
CancelProject	sends	receives		The Content Connector launches this event to cancel a project it previously submitted. This also launches events to notify the provider of the cancellation and remove any files that have been uploaded for the project.
CompletedProject	retrieves	creates		The Clay Tablet Platform launches this event after all the tasks for a project have been approved or cancelled, and it marks the project as completed.

3.8 AssetTaskUpdate Events

An `AssetTaskUpdate` is a request for an update to previously translated content.

The following diagram shows how AssetTaskUpdate events work together in sequence:


Producer Connector
Step 1: NeedAssetTaskUpdate event
Clay Tablet Platform
Step 2: StartNeedAssetTaskUpdate event
Step 3: AcceptNeedAssetTaskUpdate event
Provider Connector

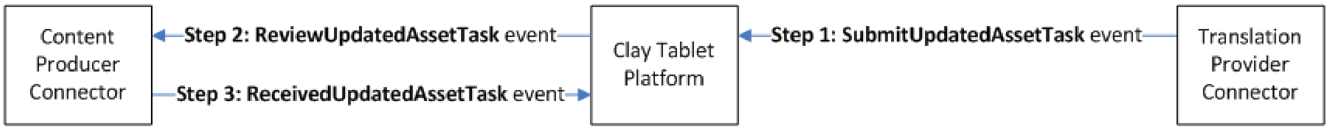
Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	NeedAssetTaskUpdate	sends	receives		The Content Connector notifies the Clay Tablet Platform that there is updated content to translate.

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
2	StartNeedAssetTaskUpdate		creates	retrieves	The Clay Tablet Platform instructs the Translation Connector to translate the updated content.
3	AcceptNeedAssetTaskUpdate		receives	sends	The Translation Connector notifies the Clay Tablet Platform that it accepts updating the translation for the content it previously translated.

3.9 UpdatedAssetTask Events

An `UpdatedAssetTask` is updated translated content.

The following diagram shows how UpdatedAssetTask events work together in sequence:

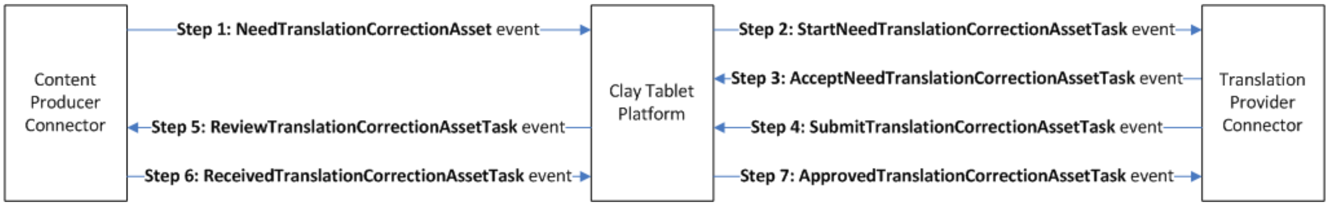

Content Producer Connector
Step 2: ReviewUpdatedAssetTask event
Step 3: ReceivedUpdatedAssetTask event
Clay Tablet Platform
Step 1: SubmitUpdatedAssetTask event
Translation Provider Connector

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	SubmitUpdatedAssetTask		receives	sends	The Translation Connector sends the updated content to the Clay Tablet Platform.
2	ReviewUpdatedAssetTask	retrieves	creates		The Clay Tablet Platform sends the updated translated content to the Content Connector for review.
3	ReceivedUpdatedAssetTask	sends	receives		The Content Connector acknowledges that it has received the updated translated content for review.

3.10 TranslationCorrectionAsset Events

A `TranslationCorrectionAsset` is the request to correct a translation, and later it is the actual corrected translation.

The following diagram shows how TranslationCorrectionAsset events work together in sequence:


Content Producer Connector
Clay Tablet Platform
Translation Provider Connector
Step 1: NeedTranslationCorrectionAsset event
Step 2: StartNeedTranslationCorrectionAssetTask event
Step 3: AcceptNeedTranslationCorrectionAssetTask event
Step 4: SubmitTranslationCorrectionAssetTask event
Step 5: ReviewTranslationCorrectionAssetTask event
Step 6: ReceivedTranslationCorrectionAssetTask event
Step 7: ApprovedTranslationCorrectionAssetTask event

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
1	NeedTranslation CorrectionAsset	sends	receives		The Content Connector informs the Clay Tablet Platform that translation correction is required.
2	StartNeedTranslation CorrectionAssetTask		creates	retrieves	The Clay Tablet Platform sends the Translation Connector the translation that requires correction.
3	AcceptNeedTranslation CorrectionAssetTask		receives	sends	The Translation Connector sends the Clay Tablet Platform its acceptance of the translation correction task.
4	SubmitTranslation CorrectionAssetTask		receives	sends	After the translation is corrected, the Translation Connector sends the Clay Tablet Platform the corrected translation for review.
5	ReviewTranslation CorrectionAssetTask	retrieves	creates		The Clay Tablet Platform sends the Content Connector the corrected translated content to review.

Step	Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
6	ReceivedTranslationCorrectionAssetTask	sends	receives		The Content Connector has reviewed and approved the translation correction.
7	ApprovedTranslationCorrectionAssetTask		creates	retrieves	The Clay Tablet Platform notifies the Translation Connector that the content provider has approved the corrected translation.

3.11 UpdateTMAsset Events

An UpdateTMAsset is translated content that the content producer sends to the translation provider so that it can update the translation memory (TM).

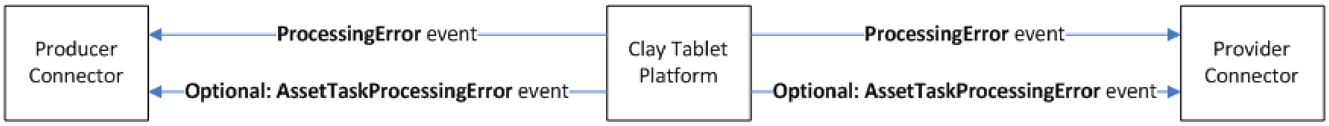
The following diagram shows how UpdateTMAsset events work together in sequence:

Step	Event	Content Connector	Platform	Translation Connector	Description
1	CreateUpdateTMAsset	sends	receives		The Content Connector sends the corrected translation content for updating the TM to the Platform.
2	StartUpdateTMAsset		creates	retrieves	The Platform sends the corrected translation content for updating the TM to the Translation Connector.
3	AcceptUpdateTMAsset		receives	sends	The Translation Connector accepts the corrected translation content for updating the TM.

3.12 ProcessingError Events

Processing errors can occur when the Clay Tablet Platform is processing an event that it receives.

The following diagram shows how ProcessingError events work together in sequence:



Event	Content Connector	Clay Tablet Platform	Translation Connector	Description
ProcessingError	retrieves	creates	retrieves	The Clay Tablet Platform sends this event to both the Content Connector and the Translation Connector when a processing error occurs.
AssetTaskProcessingError	retrieves	creates	retrieves	If the error occurs specifically to an asset task (translation), the Clay Tablet Platform sends additional information to both the Content Connector and the Translation Connector.

4 Events for Building a Content Connector

A content management system (CMS) or another content application implements a Clay Tablet *Content Connector* to:

- group multiple content items (*assets*) together, to send them to the translation provider as a single job or project. For details, see "Grouping Assets for Translation" on page 40 and "Grouping Assets into Jobs Using Job Metadata Interfaces" on page 41.
- send files for translation to the Clay Tablet Platform storage buckets and translation request messages to the Clay Tablet Platform, using events. For a list and descriptions of these events, see "Events a Content Connector Can Send" on page 42. For an example of using events to send content for translation, see "Code Sample: Sending Content for Translation" on page 67.
- retrieve translated content from the Clay Tablet Platform, using events. For details, see "Events a Content Connector Receives" on page 75. For examples of using events to retrieve translated content, see "Code Samples: Receiving Events" on page 83.

The Clay Tablet Platform uses Message Queues to provide asynchronous communications with external applications, such as a CMS. For information about asynchronous communications, see http://en.wikipedia.org/wiki/Asynchronous_communication.

Note about the XML format of assets sent for translation: The format in which you send content for translation is important. There is an industry standard called XLIFF, which we recommend you review and adhere to if possible for content sent for translation. For information about this standard, see: http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html

4.1 Grouping Assets for Translation

A producer often groups multiple assets to submit them together to the translation provider. You can group assets as a *project*, a *job*, or both.

Grouping Assets as a Project

You can use the **SubmitProject** event to group multiple assets submitted to the translation provider into a single project. In the **CreateAsset** event, you use the **setProjectId** method to specify the Project ID for each asset. You can also associate supports assets with a project using the **CreateSupportAsset** event and the **setProjectId** method.

Grouping assets as a project has the following advantage:

- A project is open-ended, because it does not include a fixed number of assets. After submitting a project, you can use the **CreateAsset** event to add additional assets by specifying the same project ID.

Grouping assets as a project has the following disadvantages:

- Since Clay Tablet's message-base architecture is asynchronous, the provider does not always receive the messages in the same order that the producer sends them. Even if the producer always sends the

SubmitProject event after sending all the **CreateAsset** events for the project, the messages may not arrive in the same order. As the result, the translation provider cannot determine when it has received all the assets in the project.

- There is very limited metadata associated with a project. The translation provider may find this metadata insufficient.

Grouping Assets as a Job

A job is identified by job ID. The number of assets in a job is specified by the asset task count.

The `IJobMetadata` interface specifies the job ID (`jobId`) and the asset task count (`AssetTaskCount`).

Grouping assets as a job has the following advantages:

- Because the number of assets in a job is specified, the translation provider can determine when it receives all the assets in a job.
- The producer can associate extensive metadata with the project, using additional interfaces.

Grouping assets as a job has the following disadvantage:

- Because the number of assets in a job is specified, the provider can not add additional assets to the job later on.

Note: If a project contains exactly one job, the project ID can be the same as the job ID.

For detailed information on grouping content for translation as a job, see "Grouping Assets into Jobs Using Job Metadata Interfaces" on page 41.

4.2 Grouping Assets into Jobs Using Job Metadata Interfaces

For a comparison of grouping assets into projects or jobs, see "Grouping Assets for Translation" on page 40.

Reference	com.claytablet.model.Event.metadata
Supported Versions	Version 3.2 and higher

Clay Tablet provides the following interfaces to support grouping assets into a job and to support a rich set of metadata about a job:

Interface	Description
IJobMetadata	Specifies the metadata for a job, including the job ID (`jobId`) and the asset task count (`AssetTaskCount`). It specifies an extensible set of additional translation-provider-specific metadata.
IMetadataGroup	Specifies the metadata for a group of assets in a job.

Interface	Description
IMetadata	Specifies the metadata for an asset in a job.
IAttribute	Specifies the names and values of metadata of assets in a job.

The IMetadataGroup, IMetadata, and IAttribute interfaces define a mechanism to extend standard set of job metadata with translation-provider-specific metadata.

The MetadataFactory class contains a group of static factory methods to create instances of job metadata-related interfaces. You use these factory methods instead of directly instantiating implementation classes of these interfaces.

4.2.1 Attaching Job Metadata to Assets

1. Use the factory methods from the MetadataFactory class to create the IJobMetadata interface, and the other interfaces, if required.
2. Set all the metadata fields.
3. In the CreateAsset and CreateSupportAsset classes, use the setJobMetadata method to attach the metadata to the assets and support assets for the job.

All assets and support assets with the same job metadata attached are part of the same job.

4.3 Events a Content Connector Can Send

Reference	com.claytablet.model.Event.producer

A Content Connector can launch the following events:

Event	Description
AcknowledgeAssetTaskTranslation	Successfully receives translated content in a single target language (asset task).
ApproveAssetTask	Reviews and approves translated content in a single target language (asset task).
ApproveAssetTaskQuote	Reviews and approves a quote for translating content into a single target language (asset task). This typically enables the translation to start.

Event	Description
CancelAsset	Cancels content it previously sent for translation (an *asset*). This also launches events to notify the translation provider of the cancellation and removes any files that were uploaded for the translation request.
CancelProject	Cancels a translation project it previously submitted. This also launches events to notify the translation provider of the cancellation and removes any files that were uploaded for the project.
CancelSupportAsset	Cancels supporting content (*support asset*) it previously sent to the translation provider. This type of content is not translated. This event also launches events to notify the translation provider of the cancellation and removes the file that was uploaded for the support asset.
CreateAsset	Sends content for translation into one or more target languages. This content is an *asset*.
CreateSupportAsset	Sends supporting content (a *support asset*) to the translation provider to provide helpful context for the translator. This type of content is not translated.
CreateUpdateTMAsset	Sends corrected translated content back to the translation provider so that it can update the translation memory (TM).
NeedAssetTaskUpdate	Sends the translation provider updated content to translate. (An earlier version of the content was previously translated.)
NeedTranslationCorrectionAsset	Sends the translation provider a request to correct translated content.
ReceivedTranslationCorrectionAssetTask	Indicates that it has reviewed and approved the translation correction.
ReceivedUpdatedAssetTask	Indicates that it has received a translation update for review.
RejectAssetTask	Rejects the translation to a single target language (an *asset task*). (For example, the translation itself is incomplete or wrong.) This event may contain notes indicating the reason for rejection and a new file might have been uploaded for the asset task that a translation provider can review.

Event	Description
RejectAssetTaskQuote	Rejects a quote for translation to a single target language (an *asset task*). This typically cancels the translation to that target language.
SubmitProject	Submits a project for processing.

4.3.1 Acknowledging Receiving Translated Content (AcknowledgeAssetTaskTranslation)

Class	AcknowledgeAssetTaskTranslation
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	AcknowledgeAssetTaskTranslation ()
Supported Versions	Version 3.3 and higher

The Content Connector launches the `AcknowledgeAssetTaskTranslation` event when it successfully receives translated content in a single target language (asset task).

Note: This is different than the content provider accepting the translation, which is the `ApproveAssetTask` event, described on page 45.

For information about asset-task-translation events, see page 34.

This class has the following methods:

Return Type	Method and Parameters
System.String	getAssetTaskId ()
void	setAssetTaskId (System.String assetTaskId)

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml
validate

4.3.2 Approving Translated Content (ApproveAssetTask)

Class	ApproveAssetTask
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ApproveAssetTask ()
Supported Versions	All versions

The Content Connector launches the `ApproveAssetTask` event after it reviews and approves translated content in a single target language (asset task).

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getReviewNote ()	
System.boolean	isWithContent ()	
System.String	RanderToXML (ApproveAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setReviewNote (System.String reviewNote)	

Return Type	Method and Parameters	Additional Information
void	setWithContent (System.boolean withContent)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

4.3.3 Approving a Translation Quote (ApproveAssetTaskQuote)

Class	ApproveAssetTaskQuote
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ApproveAssetTaskQuote ()
Supported Versions	Version 3.5 and higher

The Content Connector launches the `ApproveAssetTaskQuote` event after it reviews and approves a quote for translating content into a single target language (asset task). This typically enables the translation to start.

For information about asset-task-quote events, see page 31.

This class has the following methods:

Return Type	Method and Parameters
System.String	getApprovalNote ()

Return Type	Method and Parameters
System.String	getAssetTaskId ()
void	setApprovalNote (System.String approvalNote)
void	setAssetTaskId (System.String assetTaskId)

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	validate

4.3.4 Canceling Content Sent for Translation (CancelAsset)

Class	CancelAsset
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CancelAsset ()
Supported Versions	All versions

The Content Connector launches the `CancelAsset` event to cancel content it previously sent for translation (an *asset*). This also launches events to notify the translation provider of the cancellation and removes any files that were uploaded for the translation request.

Note: Many translation technologies *do not* support canceling content sent for translation – so be aware that these features may not work in your particular implementation.

For information about asset events, see page 26.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	

Return Type	Method and Parameters	Additional Information
System.String	RanderToXML (CancelAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.5 Canceling a Project Sent for Translation (CancelProject)

Class	CancelProject
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CancelProject ()
Supported Versions	All versions

The Content Connector launches the `CancelProject` event to cancel a translation project it previously submitted. This also launches events to notify the translation provider of the cancellation and removes any files that were uploaded for the project.

Note: Many translation technologies *do not* support canceling a project – so be aware that these features may not work in your particular implementation.

For information about project events, see page 34.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getProjectId ()	
System.String	RanderToXML (CancelProject eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setProjectId (System.String projectId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.6 Canceling Supporting Content Sent to the Translation Provider (CancelSupportAsset)

Class	CancelSupportAsset
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CancelSupportAsset ()
Supported Versions	All versions

The Content Connector launches the `CancelSupportAsset` event to cancel supporting content (*support asset*) it previously sent to the translation provider, to provide helpful context for the translator. This type of content is not translated.

This event also launches events to notify the translation provider of the cancellation and removes the file that was uploaded for the support asset.

Note: Many translation technologies *do not* support canceling supporting content – so be aware that these features may not work in your particular implementation.

For information about support-asset events, see page 27.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getSupportAssetId ()	
System.String	RanderToXML (CancelSupportAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setSupportAssetId (System.String supportAssetId)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

4.3.7 Sending Content for Translation (CreateAsset)

Class	CreateAsset
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CreateAsset ()
Supported Versions	All versions

The Content Connector launches the `CreateAsset` event to send content for translation into one or more target languages. This content is an *asset*.

Best practice: Group related assets together as a project or a job so that the provider will process the translations together. For information on considerations on deciding whether to group assets as a project or a job, see "Grouping Assets for Translation" on page 40.

Note: If a project identifier is not assigned to an asset, the system bundles the assets and assigns a project IDs to them.

- To group related assets together for processing as a single project, use the `setProjectId` method to assign a project identifier, and use the `SubmitProject` event, described in "Submitting a Project for Translation (SubmitProject)" on page 65, to describe and submit the project.
- For detailed instructions on grouping related assets as a job, see "Grouping Assets into Jobs Using Job Metadata Interfaces" on page 41.

For information about asset events, see page 26.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAdditionalServicesRequest ()	
System.String	getAssetId ()	
System.String	getAssetNativeId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.String	getProjectId ()	
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String[]	getTargetLanguageCodes ()	

Return Type	Method and Parameters	Additional Information
System.boolean	isPendingQuote ()	Supported in versions 3.5 and higher of the SDK. See the note below for additional information.
System.String	RanderToXML (CreateAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAdditionalServicesRequest (System.String additionalServicesRequest)	
void	setAssetId (System.String assetId)	
void	setAssetNativeId (System.String assetNativeId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm."FileType" fileType)	
void	setJobMetadata (IJobMetadata jobMetaData)	
void	setName (System.String name)	
void	setPendingQuote (boolean pendingQuote)	Supported in versions 3.5 and higher of the SDK. See the note below for additional information.
void	setProjectId (System.String projectId)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	

Return Type	Method and Parameters	Additional Information
void	setTags (System.String tags)	
void	setTargetLanguageCodes (System.String [] targetLanguageCodes)	
System.String ▪ If validation passes, this is null. ▪ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

Note: If the value of the `isPendingQuote` and `setPendingQuote` methods are `true`, the asset task is submitted to the translation to provide a quote, and not immediately for translation. After receiving the quote, the producer can use the `ApproveAssetTaskQuote` method to approve the quote, which instructs the provider to start the translation. Alternatively, the producer can use the `RejectAssetTaskQuote` method to reject the quote, which cancels the asset task. The default value of these methods is `false`.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.8 Sending Supporting Content to Translation (CreateSupportAsset)

Class	CreateSupportAsset
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CreateSupportAsset ()
Supported Versions	All versions

The Content Connector launches the `CreateSupportAsset` event to send supporting content *support asset* to the translation provider to provide helpful context for the translator. This type of content is not translated.

For information about support-asset events, see page 27.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.String	getProjectId ()	
System.String	getSupportAssetId ()	
System.String	getSupportAssetNativeId ()	
System.String	getTags ()	
System.String	RanderToXML (CreateSupportAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	

Return Type	Method and Parameters	Additional Information
void	setJobMetadata (IJobMetadata jobMetaData)	
void	setName (System.String name)	
void	setProjectId (System.String projectId)	
void	setSupportAssetId (System.String supportAssetId)	
void	setSupportAssetNativeId (System.String supportAssetNativeId)	
void	setTags (System.String tags)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

4.3.9 Sending Corrected Translated Content to Update the Translation Memory (CreateUpdateTMAsset)

Class	CreateUpdateTMAsset
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent

Constructor	CreateUpdateTMAsset ()
Supported Versions	Version 3.0 and higher

The Content Connector launches the `CreateUpdateTMAsset` event when the content producer sends corrected translated content back to the translation provider so that it can update the translation memory (TM).

For information about update-TM-asset events, see page 38.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetId ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getRelatedAssetTaskId ()	
System.String	getSourceLanguageCode ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (CreateUpdateTMAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	

Return Type	Method and Parameters	Additional Information
void	setJobMetadata (IJobMetadata jobMetaData)	
void	setRelatedAssetTaskId (System.String assetNativeId)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.10 Sending Content for a Translation Update (NeedAssetTaskUpdate)

Class	NeedAssetTaskUpdate
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	NeedAssetTaskUpdate ()
Supported Versions	Version 3.0 and higher

The Content Connector launches the `NeedAssetTaskUpdate` event when the content producer sends the translation provider updated content to translate. (An earlier version of the content was previously translated.)

For information about asset-task-update events, see page 35.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	RanderToXML (NeedAssetTaskUpdate eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.11 Sending Translated Content for Correction (NeedTranslationCorrectionAsset)

Class	NeedTranslationCorrectionAsset
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	NeedTranslationCorrectionAsset ()
Supported Versions	Version 3.0 and higher

The Content Connector launches the `NeedTranslationCorrectionAsset` event when the content producer asks the translation provider to correct translated content.

- If a project identifier is assigned, then the asset is not processed until the project it belongs to is submitted.
- If no project identifier is assigned, the system bundles the asset and assign a project to it.

Important: If there are known related assets that should be submitted at the same time, assign a project identifier to each asset.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	
System.String	getAssetNativeId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.String	getProjectId ()	
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (NeedTranslationCorrectionAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setAssetNativeId (System.String assetNativeId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	

Return Type	Method and Parameters	Additional Information
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm."FileType" fileType)	
void	setJobMetadata (IJobMetadata jobMetaData)	
void	setName (System.String name)	
void	setProjectId (System.String projectId)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTags (System.String tags)	
void	setTargetLanguageCodes (System.String [] targetLanguageCodes)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

4.3.12 Acknowledging Receiving Translated Content for Correction (ReceivedTranslationCorrectionAssetTask)

Class	ReceivedTranslationCorrectionAssetTask

Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ReceivedTranslationCorrectionAssetTask ()
Supported Versions	Version 3.0 and higher

The Content Connector launches the `ReceivedTranslationCorrectionAssetTask` event when the content producer has reviewed and approved the translation correction.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getReviewNote ()	
System.boolean	isWithContent ()	
System.String	RanderToXML (ReceivedTranslationCorrectionAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setReviewNote (System.String reviewNote)	
void	setWithContent (System.boolean withContent)	

Return Type	Method and Parameters	Additional Information
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

4.3.13 Acknowledging Receiving Updated Translated Content for Review (ReceivedUpdatedAssetTask)

Class	ReceivedUpdatedAssetTask
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ReceivedUpdatedAssetTask ()
Supported Versions	Version 3.0 and higher

The Content Connector launches the `ReceivedUpdatedAssetTask` event when the content producer receives a `ReviewUpdatedAssetTask` event, indicating that the translation update is ready for review.

For information about updated-asset-task events, see page 36.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	

Return Type	Method and Parameters	Additional Information
System.String	RanderToXML (ReceivedUpdatedAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
System.String ▪ If validation passes, this is null. ▪ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.14 Rejecting Translated Content (RejectAssetTask)

Class	RejectAssetTask
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	RejectAssetTask ()
Supported Versions	All versions

The Content Connector launches the `RejectAssetTask` event when it rejects the translation to a single target language (an *asset task*). (For example, the translation itself is incomplete or wrong.) This event may contain notes indicating the reason for rejection and a new file might have been uploaded for the asset task that a translation provider can review.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getReviewNote ()	
System.boolean	isWithContent ()	
System.String	RanderToXML (RejectAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setReviewNote (System.String reviewNote)	
void	setWithContent (System.boolean withContent)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.3.15 Rejecting a Translation Quote (RejectAssetTaskQuote)

Class	RejectAssetTaskQuote
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	RejectAssetTaskQuote ()
Supported Versions	Version 3.5 and higher

The Content Connector launches the `RejectAssetTaskQuote` event when it rejects a quote for translation to a single target language (an *asset task*). This typically cancels the translation to that target language.

For information about asset-task-quote events, see page 31.

This class has the following methods:

Return Type	**Method**
System.String	getAssetTaskId ()
System.String	getRejectionNote ()
void	setAssetTaskId (System.String assetTaskId)
void	setRejectionNote (System.String rejectionNote)

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml
validate

4.3.16 Submitting a Project for Translation (SubmitProject)

Class	SubmitProject
Reference	com.claytablet.model.Event.producer
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent

Constructor	SubmitProject ()
Supported Versions	All versions

The Content Connector launches the `SubmitProject` event when it submits a project for processing.

Note: The Clay Tablet Platform can also launch this event bundles together a floating set of content (assets) for processing.

For information about project events, see page 34.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAdditionalServicesRequest ()	
System.DateTime	getDeadline ()	
System.String	getDescription ()	
System.String	getName ()	
System.String	getProjectId ()	
System.String	getProjectNativeId ()	
System.String	getTags ()	
System.String	RanderToXML (SubmitProject eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAdditionalServicesRequest (System.String additionalServicesRequest)	
void	setDeadline (System.DateTime deadline)	
void	setDescription (System.String description)	
void	setName (System.String name)	
void	setProjectId (System.String projectId)	

Return Type	Method and Parameters	Additional Information
void	setProjectNativeId (System.String projectNativeId)	
void	setTags (System.String tags)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.4 Code Sample: Sending Content for Translation

The following code sample illustrates how to use the events described in this section to send content for translation.

Note: The following sample code is also included in the SDK package, in the `\Samples\ProducerSample_DotNet\ProducerSample\Sample-Sender\SendFileTest.cs` file.

```
public class SendFileTest
  {
    static void Main(string[] args)
    {
      /*
      * Sample configuration setting
      *
      * Check the app.config in the sample.
      * For web applications, configuration is in the web.config file.
      *
      * When configuring the three keys below, please use an absolute path.
      * Since Clay Tablet may be used in web applications, the absolute path
      * can be configured outside the web folder, to keep account information
      * and connectionContent information info more secure.
      *
        <appSettings>
           <add key="CTT2_SourceAccount"
                       value="E:\...\ProducerSample\accounts\source.xml" />
           <add key="CTT2_TargetAccount"
                       value="E:\...\ProducerSample\accounts\target.xml" />
           <add key="CTT2_ConnectionContext_Folder"
                       value="E:\...\ProducerSample\data\" />
           </appSettings>
      *
      *
      */

      // Load ConnectionContext. SDK looks for a file called
      // "connectionContext.xml"
      // in the folder -- "CTT2_ConnectionContext_Folder" pointed.

      // Refer to the documentation for more information.
      ConnectionContext context = new ConnectionContext();

      // Load parameters, if "connectionContext.xml" exists
      context.load();

      // Initialize a source account. SDK looks for the source account
      // file where the appsetting ("CTT2_SourceAccount") points.
      // If the SDK cannot find the appsetting ("CTT2_SourceAccount"), it
      // throws an error.
      SourceAccountProvider sap = new SourceAccountProvider();

      //Get source account
      Account sourceAccount = sap.get();

      /* If you do not want to use the appsetting ("CTT2_SourceAccount"),
      you can load the account XML file as follows:
```

```
String sourceAccountFilePath = @"C:\...\....\source.xml";

String SourceAccountXML = com.claytablet.util.FileUtil.ReadStringFromFile
(sourceAccountFilePath);

SourceAccountProvider sap = new SourceAccountProvider(SourceAccountXML);
*/

// Initialize a target account. SDK looks for the source account
// file where the appsetting ("CTT2_TargetAccount") points.
TargetAccountProvider tap = new TargetAccountProvider();

// Initialize a QueuePublisherService.
// Clay Tablet uses QueuePublisherServiceSQS.
QueuePublisherService queuePublisherService = new
            QueuePublisherServiceSQS();
queuePublisherService.setPublicKey(sourceAccount.getPublicKey());
queuePublisherService.setPrivateKey(sourceAccount.getPrivateKey());
queuePublisherService.setEndpoint(sourceAccount.getQueueEndpoint());

// Initialize a StorageClientService.
// Clay Tablet uses StorageClientServiceS3.
StorageClientService storageClientService = new StorageClientServiceS3();
storageClientService.setPublicKey(sourceAccount.getPublicKey());
storageClientService.setPrivateKey(sourceAccount.getPrivateKey());
storageClientService.setStorageBucket(sourceAccount.getStorageBucket());

// Initialize a ProducerSender.
ProducerSender sender = new ProducerSender(context, sap, tap,
            queuePublisherService, storageClientService);

// Create a new project ID.
String projectId = IdGenerator.createId();

// Create an IJobMetadata interface to group all the assets into one job.
IJobMetadata  jobMetadata  =  JobMetadataFactory.newJobMetadata();
```

```
// One job per project.
jobMetadata.setJobId(projectId);
jobMetadata.setJobName("jobName");

// 4 assets, 2 target languages per asset, so there are 8 asset tasks.
jobMetadata.setAssetTaskCount(8);
jobMetadata.setProducerId("test-producerId");
jobMetadata.setPoReference("poReference");
jobMetadata.setDescription("description");
jobMetadata.setClient("ClayTablet");
jobMetadata.setCreator("foo");
jobMetadata.setCreatorContact("foo@clay-tablet.com");
jobMetadata.setStartDate("01/01/2011");
jobMetadata.setDueDate("01/01/2012");

DirectoryInfo curDir = (new DirectoryInfo(".")).Parent.Parent;
String testFilesPath = curDir.FullName + "\\TestFiles\\";

try
{
  // Send test asset1 file.
  Console.WriteLine("Send first asset...");

  // Create the first asset.
  CreateAsset asset1 = new CreateAsset();
  asset1.setAssetId(IdGenerator.createId());
  asset1.setProjectId(projectId);
  asset1.setAssetNativeId("your reference here");
  asset1.setName("First Asset");
  asset1.setTags("first, asset");
  asset1.setDescription("First Asset Description.");

  // Set content Type
  asset1.setContentType(ContentType.Technical);
  // Set file Type
  asset1.setFileType(FileType.Text);
  // Set file extension
```

```
asset1.setFileExt("txt");

// Set asset source language.
asset1.setSourceLanguageCode(Language.English_US);

// Set asset target languages.
String[] targetLanguageCodes = new String[2];
targetLanguageCodes[0] = Language.French;
targetLanguageCodes[1] = Language.Spanish;
asset1.setTargetLanguageCodes(targetLanguageCodes);

asset1.setAdditionalServicesRequest("First Asset –
            Additional Services Request.");

// Attach job metadata.
asset1.setJobMetadata(jobMetadata);

// Send the event.
sender.sendEvent(asset1, testFilesPath + "asset1.txt");
Console.WriteLine("Send first asset --- OK\n");

// Create the second asset.
Console.WriteLine("Send second asset...");
CreateAsset asset2 = new CreateAsset();
asset2.setAssetId(IdGenerator.createId());
asset2.setProjectId(projectId);
asset2.setAssetNativeId("asset2");
asset2.setName("Second Asset");
asset2.setTags("second, asset");
asset2.setDescription("Second Asset Description.");
asset2.setContentType(ContentType.Technical);
asset2.setFileType(FileType.PDF);
asset2.setFileExt("pdf");
asset2.setSourceLanguageCode(Language.English_US);
targetLanguageCodes = new String[2];
targetLanguageCodes[0] = Language.French;
targetLanguageCodes[1] = Language.Spanish;
```

```
asset2.setTargetLanguageCodes(targetLanguageCodes);
asset2.setAdditionalServicesRequest("Second Asset –
            Additional Services Request.");

// Attach job metadata.
asset2.setJobMetadata(jobMetadata);

// Send the event.
sender.sendEvent(asset2, testFilesPath + "asset2.pdf");
Console.WriteLine("Send second asset --- OK\n");

// Create the third asset.
Console.WriteLine("Send third asset...");
CreateAsset asset3 = new CreateAsset();
asset3.setAssetId(IdGenerator.createId());
asset3.setProjectId(projectId);
asset3.setAssetNativeId("asset3");
asset3.setName("Third Asset");
asset3.setTags("Third, asset");
asset3.setDescription("Third Asset Description.");
asset3.setContentType(ContentType.Technical);
asset3.setFileType(FileType.HTML);
asset3.setFileExt("html");
asset3.setSourceLanguageCode(Language.English_US);
targetLanguageCodes = new String[2];
targetLanguageCodes[0] = Language.Spanish;
targetLanguageCodes[1] = Language.German;
asset3.setTargetLanguageCodes(targetLanguageCodes);
asset3.setAdditionalServicesRequest("Third Asset - Additional
            Services Request.");
// Attach job metadata.
asset3.setJobMetadata(jobMetadata);

// Send the event.
sender.sendEvent(asset3, testFilesPath + "asset3.html");
Console.WriteLine("Send third asset --- OK\n");
```

```
// Create the fourth asset.
Console.WriteLine("Send fourth asset...");
CreateAsset asset4 = new CreateAsset();
asset4.setAssetId(IdGenerator.createId());
asset4.setProjectId(projectId);
asset4.setAssetNativeId("asset4");
asset4.setName("Fourth Asset");
asset4.setTags("fourth, asset");
asset4.setDescription("Fourth Asset Description.");
asset4.setContentType(ContentType.Technical);
asset4.setFileType(FileType.XML);
asset4.setFileExt("xml");
asset4.setSourceLanguageCode(Language.English_US);
targetLanguageCodes = new String[2];
targetLanguageCodes[0] = Language.French;
targetLanguageCodes[1] = Language.German;
asset4.setTargetLanguageCodes(targetLanguageCodes);
asset4.setAdditionalServicesRequest("Fourth Asset - Additional
            Services Request.");
// Attach job metadata.
asset4.setJobMetadata(jobMetadata);

// Send the event.
sender.sendEvent(asset4, testFilesPath + "asset4.xml");
Console.WriteLine("Send fourth asset --- OK\n");

// Create the first support asset.
Console.WriteLine("Send first support asset...");
CreateSupportAsset supportAsset1 = new CreateSupportAsset();
supportAsset1.setProjectId(projectId);
supportAsset1.setSupportAssetNativeId("supportAsset1");
supportAsset1.setName("First Support Asset");
supportAsset1.setTags("first, support, asset");
supportAsset1.setDescription("First Support Asset Description");
supportAsset1.setContentType(ContentType.Technical);
supportAsset1.setFileType(FileType.Text);
supportAsset1.setFileExt("txt");
```

```
// Attach job metadata.
supportAsset1.setJobMetadata(jobMetadata);

// Send the event.
sender.sendEvent(supportAsset1, testFilesPath + "supportAsset1.txt");
Console.WriteLine("Send first support asset --- OK\n");

// Create the second support asset.
Console.WriteLine("Send second support asset...");
CreateSupportAsset supportAsset2 = new CreateSupportAsset();
supportAsset2.setProjectId(projectId);
supportAsset2.setSupportAssetNativeId("supportAsset2");
supportAsset2.setName("Second Support Asset");
supportAsset2.setTags("second, support, asset");
supportAsset2.setDescription("Second Support Asset Description");
supportAsset2.setContentType(ContentType.Technical);
supportAsset2.setFileType(FileType.Other);
supportAsset2.setFileExt("dat");

// Attach job metadata.
supportAsset2.setJobMetadata(jobMetadata);

// Send the event.
sender.sendEvent(supportAsset2, testFilesPath + "supportAsset2.dat");
Console.WriteLine("Send second support asset --- OK\n");

// Submit the project.
Console.WriteLine("Submit project...");
SubmitProject project = new SubmitProject();

// Ensure the generated project ID is already assigned to all the assets.
project.setProjectId(projectId);
project.setProjectNativeId("project-submission-test-net");
project.setName("Project Submission Test");
project.setTags("project, submission, test");
project.setDescription("Project Submission Test Description.");
```

```
        project.setAdditionalServicesRequest("Project Submission
                     Test - Additional Services Request.");

        // Send the event.
        sender.sendEvent(project);
        Console.WriteLine("Submit project --- OK\n");

      }
      catch (Exception e)
      {
        Console.WriteLine("Error throws.");
        Console.WriteLine("Error Info:\n" + e.Message);
      }
      Console.WriteLine("Press any key to exit...");
      Console.ReadLine();
    }
  }
}
```

4.5 Events a Content Connector Receives

Reference	com.claytablet.model.Event.platform

A Content Connector can receive the following events from the Clay Tablet Platform:

Event	Description
CompletedProject	Receives notification that the project is marked as completed, because all the project tasks were either approved or canceled.
ReviewAssetTask	Receives notification that translated content is ready for review.
ReviewTranslationCorrectionAssetTask	Receives notification that the translation correction is ready for review.
ReviewUpdatedAssetTask	Receives notification that the translation update is ready for review.
UpdatedAssetTaskState	Receives notification that the translation provider has updated the state and/or the estimated arrival time of translated content to a particular language (an *asset task*).

Note: A Content Connector also receives the ProcessingError and AssetTaskProcessingError events, described in "Events that All Connectors Receive" on page 151.

4.5.1 Receiving Notification of a Completed Project (CompletedProject)

Class	CompletedProject
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CompletedProject ()
Supported Versions	All versions

The Clay Tablet Platform launches the CompletedProject event and sends it to the Content Connector. It launches this event when it marks the project as completed, because all the project tasks were either approved or canceled.

Recommendation: After the Content Connector receives this event, it can call the CMS to flag the project or bundle as completed or ready for the next workflow step.

For information about project events, see page 34.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getProjectId ()	
System.String	getProjectNativeId ()	
System.String	RanderToXML (CompletedProject eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setProjectId (System.String projectId)	
void	setProjectNativeId (System.String projectNativeId)	

Return Type	Method and Parameters	Additional Information
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

4.5.2 Receiving an Instruction to Review Translated Content (ReviewAssetTask)

Class	ReviewAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ReviewAssetTask ()
Supported Versions	All versions

The Clay Tablet Platform launches the `ReviewAssetTask` event and sends it to the Content Connector. It launches this event after a translation provider submits translated content back to the platform, indicating that it is ready for review.

A user of the Content Connector, such as a content editor, can then review the translation, often as a workflow step in the CMS. The user can then either approve or reject the translation.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	
System.String	getAssetNativeId ()	

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (ReviewAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setAssetNativeId (System.String assetNativeId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.5.3 Receiving an Instruction to Review Corrected Translated Content (ReviewTranslationCorrectionAssetTask)

Class	ReviewTranslationCorrectionAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ReviewTranslationCorrectionAssetTask ()
Supported Versions	Version 3.0 and higher

The Clay Tablet Platform launches the `ReviewTranslationCorrectionAssetTask` event and sends it to the Content Connector. It launches this event after the translation provider completes the translation correction, and the Translation Connector submits it using the `SubmitTranslationCorrectionAssetTask` event. The content producer can accept or reject this correction.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetId ()	
System.String	getAssetNativeId ()	
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (ReviewTranslationCorrectionAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setAssetNativeId (System.String assetNativeId)	
void	setAssetTaskId (System.String assetTaskId)	

Return Type	Method and Parameters	Additional Information
void	setFileExt (System.String fileExt)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.5.4 Receiving an Instruction to Review Updated Translated Content (ReviewUpdatedAssetTask)

Class	ReviewUpdatedAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ReviewUpdatedAssetTask ()
Supported Versions	Version 3.0 and higher

The Clay Tablet Platform launches the `ReviewUpdatedAssetTask` event and sends it to the Content Connector. It launches this event when the translation provider submits an update of translated content back to the content producer for review. The content producer can then accept or reject this update.

For information about updated-asset-task events, see page 36.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	
System.String	getAssetNativeId ()	
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (ReviewUpdatedAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setAssetNativeId (System.String assetNativeId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.5.5 Receiving an Update about Translation Status (UpdatedAssetTaskState)

Class	UpdatedAssetTaskState
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	UpdatedAssetTaskState ()
Supported Versions	All versions

The Clay Tablet Platform launches the `UpdatedAssetTaskState` and sends it to the Content Connector. It launches this event when the translation provider has updated the state and/or the estimated arrival time of translated content to a particular language (an *asset task*).

For information about asset-task-state events, see page 33.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetId ()	
System.String	getAssetNativeId ()	
System.String	getAssetTaskId ()	
System.DateTime	getEta ()	
System.String	getNativeState ()	
System.String	getNotes ()	
System.String	getProjectId ()	
int	getTranslationPercentage ()	
System.String	RanderToXML (UpdatedAssetTaskState eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.

Return Type	Method and Parameters	Additional Information
void	setAssetId (System.String assetId)	
void	setAssetNativeId (System.String assetNativeId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setEta (System.DateTime eta)	
void	setNativeState (System.String nativeState)	
void	setNotes (System.String notes)	
void	setProjectId (System.String projectId)	
void	setTranslationPercentage (int translationPercentage)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

4.6 Code Samples: Receiving Events

This section contains the following examples of the Content Connector receiving events:

- "Code Sample: Handling Errors and Receiving Translated Files" on page 84
- "Code Sample: Receiving a Message, Deserializing an Event, and Calling Your Own Event Receiver to Handle the Event" on page 87

Note: All sample code is also included in the `Samples` folder in the SDK package.

4.6.1 Code Sample: Handling Errors and Receiving Translated Files

The following code sample illustrates how the Content Connector uses the events described in this section to handle errors and receive translated files.

Note: The following sample code is also included in the SDK package, in the `\Samples\ProducerSample_DotNet\ProducerSample\Sample-Receiver\MyProducerReceiver.cs` file.

```
public class MyProducerReceiver
{
  private ProducerSender sender;
  private StorageClientService storageClientService;
  private ConnectionContext context;

  public MyProducerReceiver(ConnectionContext context,
              StorageClientService storageClientService, ProducerSender sender)
  {
    // Pass a ConnectionContext to retrieve the path where
    // the translated files will be saved.

    // For more information, refer to the documentation.
    this.context = context;

    // Pass a StorageClientService to download translated files.
    this.storageClientService = storageClientService;

    // Pass a ProducerSender to send events and files, if needed.
    // Refer to the Sample-Sender folder for examples of sending events and files.
    this.sender = sender;
  }

  // Event handling
  public HandleResult ReceiveEvent(IEvent cttEvent)
```

```
{

  if (cttEvent is CompletedProject)
    return HandleEvent((CompletedProject)cttEvent);
  else if (cttEvent is UpdatedAssetTaskState)
    return HandleEvent((UpdatedAssetTaskState)cttEvent);
  else if (cttEvent is ProcessingError)
    return HandleEvent((ProcessingError)cttEvent);
  else if (cttEvent is ReviewAssetTask)
    return HandleEvent((ReviewAssetTask)cttEvent);
  else
  {
    HandleResult handleResult = new HandleResult();
    handleResult.Success = false;
    handleResult.CanDeleteMessage = true;
    handleResult.ErrorMessage = "Can't identify the Event Type.";

    return handleResult;
  }

}

private HandleResult HandleEvent(CompletedProject cttEvent)
{
  HandleResult handleResult = new HandleResult();
  // Insert producer integration code here.
  // For example, mark the local project as completed in the CMS.
  return handleResult;

}
private HandleResult HandleEvent(UpdatedAssetTaskState cttEvent)
{
  HandleResult handleResult = new HandleResult();
  // Insert producer integration code here.
  // For example, mark the local project as completed in the CMS.
  return handleResult;
}
```

```
  private HandleResult HandleEvent(ProcessingError cttEvent)
  {
    HandleResult handleResult = new HandleResult();
    // Insert producer integration code here.
    // An error occured. Examine and take appropriate action.

    return handleResult;
  }

  private HandleResult HandleEvent(ReviewAssetTask cttEvent)
  {
    HandleResult handleResult = new HandleResult();
    // Insert producer integration code here.
    // For example, send the asset to the CMS and mark it for review.
    /* useful code here

    String savePath = context.getTargetDirectory();

    // Download the translated file to savePath, and return the file path,
    // located in local file system.

    String downloadFilePath =
      storageClientService.downloadLatestAssetTaskVersion(
        cttEvent.getAssetTaskId(),savePath);

    // Read the content from the file, and return it to the CMS ...
    *
    */

    return handleResult;

  }

}
```

4.6.2 Code Sample: Receiving a Message, Deserializing an Event, and Calling Your Own Event Receiver to Handle the Event

The following code sample illustrates how the Content Connector uses the events described in this section to receive a message, deserialize an event, and calling your own event receiver to handle the event.

Note: The following sample code is also included in the SDK package, in the `\Samples\ProducerSample_DotNet\ProducerSample\Sample-Receiver\ReceiveFileTest.cs` file.

```
public class ReceiveFileTest
{

  public static void Main(string[] args)
  {

    SourceAccountProvider sap;
    TargetAccountProvider tap;

    QueueSubscriberService queueSubscriberService;
    QueuePublisherService queuePublisherService;
    StorageClientService storageClientService;
    MyProducerReceiver receiver;

    ConnectionContext context;
    ProducerSender sender;

    /*
    * Sample configuration setting
    *
    * Check the app.config in the sample.
    * For web applications, the configuration is in the web.config file.
    *
    * When configuring the three keys below, please use an absolute path.
    * Since Clay Tablet may be used in web applications, the absolute path
    * can be figured outside the web folder, to keep account information
    * and connectionContent information more secure.
    *
      <appSettings>
```

```
     <add key="CTT2_SourceAccount"
                 value="E:\...\ProducerSample\accounts\source.xml" />
     <add key="CTT2_TargetAccount"
                 value="E:\...\ProducerSample\accounts\target.xml" />
     <add key="CTT2_ConnectionContext_Folder"
                 value="E:\...\ProducerSample\data\" />
  </appSettings>
*
*
*/

// Load ConnectionContext so that SDK looks for the
// "connectionContext.xml" file
// in the folder specified by "CTT2_ConnectionContext_Folder".

// For more information, refer to the documentation.
context = new ConnectionContext();
context.load();

// Initialize a source account so that the SDK looks for the source account
// file, which the appsetting ("CTT2_SourceAccount") specifies.
sap = new SourceAccountProvider();

/* If you do not want to use appsetting ("CTT2_SourceAccount"),
You can load the account XML file as follows:

String sourceAccountFilePath = @"C:\...\....\source.xml";
String SourceAccountXML = com.claytablet.util.FileUtil.ReadStringFromFile
(sourceAccountFilePath);
SourceAccountProvider sap = new SourceAccountProvider(SourceAccountXML);
*/

// Initialize a target account so that the SDK looks for the source account
// file, which the appsetting ("CTT2_TargetAccount") specifies.
tap = new TargetAccountProvider();

Account sourceAccount = sap.get();
```

```
/* If you do not want to use appsetting ("CTT2_ TargetAccount "),
you can load the account XML file as follows:

String targetAccountFilePath = @"C:\...\....\target.xml";
String TargetAccountXML = com.claytablet.util.FileUtil.ReadStringFromFile
(targetAccountFilePath);
TargetAccountProvider tap = new TargetAccountProvider (TargetAccountXML);
*/
storageClientService = new StorageClientServiceS3();
storageClientService.setPublicKey(sourceAccount.getPublicKey());
storageClientService.setPrivateKey(sourceAccount.getPrivateKey());
storageClientService.setStorageBucket(
            sourceAccount.getStorageBucket());

queuePublisherService = new QueuePublisherServiceSQS();
queuePublisherService.setPublicKey(sourceAccount.getPublicKey());
queuePublisherService.setPrivateKey(sourceAccount.getPrivateKey());
queuePublisherService.setEndpoint(sourceAccount.getQueueEndpoint());

queueSubscriberService = new QueueSubscriberServiceSQS();
queueSubscriberService.setPublicKey(sourceAccount.getPublicKey());
queueSubscriberService.setPrivateKey(sourceAccount.getPrivateKey());
queueSubscriberService.setEndpoint(sourceAccount.getQueueEndpoint());

// Initialize a ProducerSender, which your Producer Receiver needs to send
// events.
sender = new ProducerSender(context, sap, tap,
            queuePublisherService, storageClientService);

//Initialize your own ProducerSender.
receiver = new MyProducerReceiver(context, storageClientService, sender);

Console.WriteLine(string.Format("Check SQS message for Translated Item.
            Start at {0}\t", System.DateTime.Now.ToString()) );

//Retrieve all available messages  --- if pass 0,
//or you can pass 100 if you wantto handle only 100 messages.
```

```
int maxMessages = 0;

Message[] messages = queueSubscriberService.receiveMessages(maxMessages);

Console.WriteLine("Found " + messages.Length + " messages." );

if (messages.Length > 0)
{
  int msg_Count = 0;
  foreach (Message message in messages)
  {
    msg_Count++;
    Console.WriteLine(msg_Count.ToString() +
                 ")Message Body:\n" + message.getBody() );

    try
    {
      // Deserializing, from XML to AbsEvent,
      IEvent curEvent = AbsEvent.fromXml(message.getBody());

      // Call your ProducerReceiver to handle the event.
      HandleResult curHandleResult = receiver.ReceiveEvent(curEvent);

      if (curHandleResult.CanDeleteMessage)
      {
        // Event handled, delete it from the queue.
        queueSubscriberService.deleteMessage(message);
        Console.WriteLine("Message handled, so delete it from queue." );
      }

      if (!curHandleResult.Success)
      {
        Console.WriteLine("Event handling error.\nError
                   Message:" + curHandleResult.ErrorMessage );
      }
```

```
        }
        catch (Exception e)
        {
          Console.WriteLine(e.Message );
        }


      }


    }


    Console.WriteLine("Press any key to exit...");
    Console.ReadLine();


  }
}
```

5 Events for Building a Translation Connector

A translation management system (TMS) or machine translation (MT) service implements a Clay Tablet *Translation Connector* to:

- receive translation requests and download files for translation from the Clay Tablet Platform, using events. For details, see "Events a Translation Connector Receives" on page 92. For an example of using events to receive requests and files for translation, see "Code Sample: Retrieving Messages, Deserializing Event Objects, and Handling Events" on page 128.
- retrieve metadata about a translation job. For details, see "Retrieving Job Metadata" on page 132.
- use event to send translated content to the Clay Tablet Platform, where it can be retrieved by the CMS. For details, see "Events a Translation Connector Can Send" on page 133. For examples of using events to send status updates and translated content, see "Code Sample: Sending Translation Status or Translated Files" on page 145.

The Clay Tablet Platform uses Message Queues to provide asynchronous communications with external applications, such as a TMS or an MT service. For information about asynchronous communications, see http://en.wikipedia.org/wiki/Asynchronous_communication.

Note about the XML format of assets sent for translation: The format in which you send content for translation is important. There is an industry standard called XLIFF, which we recommend you review and adhere to if possible for content sent for translation. For information about this standard, see: http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html

5.1 Events a Translation Connector Receives

Reference	com.claytablet.model.Event.platform

A Translation Connector can receive the following events from the Clay Tablet Platform:

Event	Description
AcknowledgedAssetTaskTra	Receives notification that the content producer acknowledges that it has successfully received the translation delivered back to it.
ApprovedAssetTask	Receives notification that the Content Connector reviewed and approved content translated into a particular target language (an *asset task*).
ApprovedAssetTaskQuote	Receives notification that the Content Connector approves the quote it received to translate content into a particular target language. This event instructs the Translation Connector to start the specified translation task (an *asset task*).

Event	Description
ApprovedTranslationCorrectionAssetTask	Receives notification that the Content Connector approves the translation correction.
CanceledAssetTask	Receives notification that the Content Connector cancels a translation task (*asset task*).
CanceledSupportAsset	Receives notification that the Content Connector cancels supporting content (*support asset*) it sent to the content provider. This type of content is not translated.
RejectedAssetTask	Receives notification that the Content Connector rejects the translated content for a particular target language (*asset task*). This event may contain notes indicating the reason for the rejection.
RejectedAssetTaskQuote	Receives notification that the Content Connector rejects the quote for translating content into a particular target language (*asset task*). This event also cancels this translation.
StartAssetTask	Receives notification that the Content Connector has sent content for translation into a specified target language.
StartAssetTaskQuote	Receives notification that the Content Connector requests a quote for translating content it sends to the translation provider.
StartNeedAssetTaskUpdate	Receives notification that the Content Connector sends updated content to translate.
StartNeedTranslationCorrectionAssetTask	Receives notification that the Translation Connector requires a translation correction.
StartSupportAsset	Receives notification that the Translation Connector sends supporting content (a *support asset*) to the translation into provider. This type of content is not translated.
StartUpdateTMAsset	Receives notification that the Translation Connector sends corrected translated content to the translation provider. This event instructs the translation provider to update the translation memory with this corrected translated content.

Note: A Content Connector also receives the `ProcessingError` and `AssetTaskProcessingError` events, described in "Events that All Connectors Receive" on page 151.

5.1.1 Receiving Notification that the Producer Acknowledged the Translation (AcknowledgedAssetTaskTranslation)

Class	AcknowledgedAssetTaskTranslation
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.platform.AbsAssetTaskEvent
Implements	IEvent
Constructor	AcknowledgedAssetTaskTranslation ()
Supported Versions	Version 3.3 and higher

The Clay Tablet Platform launches the `AcknowledgedAssetTaskTranslation` event and sends it to the Translation Connector. It launches this event when a content producer acknowledges that it has successfully received the translation delivered back to it.

Note: Receiving this event does not indicate that the content producer *approves* the translation. The Translation Connector may later receive either the `ApprovedAssetTask` or the `RejectedAssetTask` event, indicating whether the producer eventually approves the translation.

For information about asset-task-translation events, see page 34.

This class inherits the following methods from `com.claytablet.model.Event.platform.AbsAssetTaskEvent`, its parent abstract class, which cannot be instantiated:

Return Type	Method and Parameters
System.String	getAssetId ()
System.String	getAssetTaskId ()
System.String	getName ()
System.String	getProjectId ()
System.String	getProjectName ()
System.String	getSourceLanguageCode ()

Return Type	Method and Parameters
System.String	getTargetLanguageCode ()
void	setAssetId (System.String assetId)
void	setAssetTaskId (System.String assetTaskId)
void	setName (System.String name)
void	setProjectId (System.String projectId)
void	setProjectName (System.String projectName)
void	setSourceLanguageCode (System.String sourceLanguageCode)
void	setTargetLanguageCode (System.String targetLanguageCode)

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml
validate

5.1.2 Receiving Notification that the Producer Approved Translated Content (ApprovedAssetTask)

Class	ApprovedAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ApprovedAssetTask ()
Supported Versions	All versions

The Clay Tablet Platform launches the `ApprovedAssetTask` event and sends it to the Translation Connector. It launches this event when the Content Connector reviewed and approved content translated into a particular target language (an *asset task*).

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getReviewNote ()	
System.boolean	isWithContent ()	
System.String	RanderToXML (ApprovedAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setReviewNote (System.String reviewNote)	
void	setWithContent (System.boolean withContent)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.3 Receiving Notification that the Producer Approved Your Quote (ApprovedAssetTaskQuote)

Class	StartAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.platform.StartAssetTaskBase
Implements	IEvent
Constructor	ApprovedAssetTaskQuote ()
Supported Versions	Version 3.5 and higher

The Clay Tablet Platform launches the `StartAssetTask` event and sends it to the Translation Connector. It launches this event when the content producer approves the quote it received to translate content into a particular target language. This event instructs the Translation Connector to start the specified translation.

Note: If the Translation Connector does not support quoting, it should launch an AssetTaskProviderError event message to notify the content producer.

Tip: One of the asset task's properties is a project identifier, which associates related tasks. Use the `getProjectId` method, described below, to obtain a more complete context for the work to be performed.

For information about asset-task-quote events, see page 31.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getApprovalNote ()	
System.boolean	isPendingQuote ()	Specified by `isPendingQuote` in the StartAssetTaskBase class.
void	setApprovalNote (System.String approvalNote)	

This class inherits the following methods from `com.claytablet.model.Event.platform.StartAssetTaskBase`, its parent abstract class, which cannot be instantiated:

Return Type	Method and Parameters	Additional Information
System.String	getAdditionalServicesRequest ()	

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	
System.String	getAssetTaskId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.DateTime	getProjectDeadline ()	
System.String	getProjectId ()	
System.String	getProjectName ()	
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String	getTargetLanguageCode ()	
void	setAdditionalServicesRequest (System.String additionalServicesRequest)	
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	

Return Type	Method and Parameters	Additional Information
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	
void	setName (System.String name)	
void	setProjectDeadline (System.DateTime projectDeadline)	
void	setProjectId (System.String projectId)	
void	setProjectName (System.String projectName)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTags (System.String tags)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.4 Receiving Notification that the Producer Approved the Corrected Translated Content (ApprovedTranslationCorrectionAssetTask)

Class	ApprovedTranslationCorrectionAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	ApprovedTranslationCorrectionAssetTask ()
Supported Versions	Version 3.0 and higher

The Clay Tablet Platform launches the `ApprovedTranslationCorrectionAssetTask` event and sends it to the Translation Connector. It launches this event after the content producer reviews and approves the translation correction, using the `ReceivedTranslationCorrectionAssetTask` event.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getReviewNote ()	
System.boolean	isWithContent ()	
System.String	RanderToXML (ApprovedTranslationCorrectionAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setReviewNote (System.String reviewNote)	
void	setWithContent (System.boolean withContent)	

Return Type	Method and Parameters	Additional Information
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

5.1.5 Receiving Notification that Content for Translation Was Canceled (CanceledAssetTask)

Class	CanceledAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CanceledAssetTask ()
Supported Versions	All versions

The Clay Tablet Platform launches the `CanceledAssetTask` event and sends it to the Translation Connector. It launches this event after the content producer cancels a translation task (*asset task*).

Note: Most translation management systems (TMS) *do not* support canceling tasks, and therefore few, if any, CMS Connectors send these events to the TMS. If your system *does* support this functionality, please contact Clay Tablet Support about implementing these features. For details, see "How to Contact Clay Tablet Support" on page 12.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	RanderToXML (CanceledAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.6 Receiving Notification that Supporting Content Was Canceled (CanceledSupportAsset)

Class	CanceledSupportAsset
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	CanceledSupportAsset ()
Supported Versions	All versions

The Clay Tablet Platform launches the `CanceledSupportAsset` event and sends it to the Translation Connector. It launches this event after the content producer cancels supporting content (*support asset*) it sent to the content provider. This type of content is not translated.

Note: Most translation management systems (TMS) *do not* support canceling tasks, and therefore few, if any, CMS Connectors send these events to the TMS. If your system *does* support this functionality, please contact Clay Tablet Support about implementing these features. For details, see "How to Contact Clay Tablet Support" on page 12.

For information about support-asset events, see page 27.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getSupportAssetId ()	
System.String	RanderToXML (CanceledSupportAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setSupportAssetId (System.String supportAssetId)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.7 Receiving Notification that the Producer Rejected Translated Content (RejectedAssetTask)

Class	RejectedAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent

Constructor	RejectedAssetTask ()
Supported Versions	All versions

The Clay Tablet Platform launches the `RejectedAssetTask` event and sends it to the Translation Connector. It launches this event after the content producer rejects the translated content for a particular target language (*asset task*). This event may contain notes indicating the reason for the rejection. The content producer may have uploaded a new file for this translation that the translation provider can review.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getReviewNote ()	
System.Boolean	isWithContent ()	
System.String	RanderToXML (RejectedAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setReviewNote (System.String reviewNote)	
void	setWithContent (System.Boolean withContent)	

Return Type	Method and Parameters	Additional Information
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

5.1.8 Receiving Notification that the Producer Rejected Your Quote (RejectedAssetTaskQuote)

Class	RejectedAssetTaskQuote
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.platform.AbsAssetTaskEvent
Implements	IEvent
Constructor	RejectedAssetTaskQuote ()
Supported Versions	Version 3.5 and higher

The Clay Tablet Platform launches the `RejectedAssetTaskQuote` event and sends it to the Translation Connector. It launches this event after the content producer rejects the quote for translating content into a particular target language (*asset task*). This event also cancels this translation.

Note: If the Translation Connector does not support quoting, it should launch an AssetTaskProviderError event message to notify the content producer.

For information about asset-task-quote events, see page 31.

This class has the following methods:

Return Type	Method
System.String	getRejectionNote ()

Return Type	Method
System.String	setRejectionNote (System.String rejectionNote)

This class inherits the following methods from `class com.claytablet.model.Event.platform.AbsAssetTaskEvent`, its parent abstract class, which cannot be instantiated:

Return Type	Method and Parameters
System.String	getAssetId ()
System.String	getAssetTaskId ()
System.String	getName ()
System.String	getProjectId ()
System.String	getProjectName ()
System.String	getSourceLanguageCode ()
System.String	getTargetLanguageCode ()
void	setAssetId (System.String assetId)
void	setAssetTaskId (System.String assetTaskId)
void	setName (System.String name)
void	setProjectId (System.String projectId)
void	setProjectName (System.String projectName)
void	setSourceLanguageCode (System.String sourceLanguageCode)
void	setTargetLanguageCode (System.String targetLanguageCode)

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml
validate

5.1.9 Receiving New Content to Translate (StartAssetTask)

Class	StartAssetTask

Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.platform.StartAssetTaskBase
Implements	IEvent
Constructor	StartAssetTask ()
Supported Versions	All versions

The Clay Tablet Platform launches the `StartAssetTask` event and sends it to the Translation Connector. It launches this event after the content producer sends content for translation into specified target languages. The translation into each target language is a separate *asset task*. This event instructs the Translation Connector to start the translation.

Tip: One of the asset task's properties is a project identifier, which associates related tasks. Use the `getProjectId` method, described below, to obtain a more complete context for the work to be performed.

For information about asset-task events, see page 28.

This class has the following method:

Return Type	Method	Additional Information
boolean	isPendingQuote ()	Specified by `isPendingQuote` in the StartAssetTaskBase class.
System.String	RanderToXML (StartAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.

This class inherits the following methods from com.claytablet.model.Event.platform.StartAssetTaskBase, its parent abstract class, which cannot be instantiated:

Return Type	Method and Parameters	Additional Information
System.String	getAdditionalServicesRequest ()	
System.String	getAssetId ()	
System.String	getAssetTaskId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	

Return Type	Method and Parameters	Additional Information
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.DateTime	getProjectDeadline ()	
System.String	getProjectId ()	
System.String	getProjectName ()	
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String	getTargetLanguageCode ()	
void	setAdditionalServicesRequest (System.String additionalServicesRequest)	
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	
void	setName (System.String name)	
void	setProjectDeadline (System.DateTime projectDeadline)	

Return Type	Method and Parameters	Additional Information
void	setProjectId (System.String projectId)	
void	setProjectName (System.String projectName)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTags (System.String tags)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.10 Receiving a Request for a Translation Quote (StartAssetTaskQuote)

Class	StartAssetTaskQuote
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.platform.StartAssetTaskBase
Implements	IEvent
Constructor	StartAssetTaskQuote ()
Supported Versions	Version 3.5 and higher

The Clay Tablet Platform launches the `StartAssetTaskQuote` event and sends it to the Translation Connector. It launches this event after the content producer requests a quote for translating content it sends to the translation provider. This event instructs the Translation Connector to start the quote.

Note: If the Translation Connector does not support quoting, it should launch an AssetTaskProviderError event message to notify the content producer.

Tip: One of the asset task's properties is a project identifier, which associates related tasks. Use the `getProjectId` method, described below, to obtain a more complete context for the work to be performed.

For information about asset-task-quote events, see page 31.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System. boolean	isPendingQuote ()	Specified by `isPendingQuote` in the StartAssetTaskBase class.

This class inherits the following methods from `com.claytablet.model.Event.platform.StartAssetTaskBase`, its parent abstract class, which cannot be instantiated:

Return Type	Method and Parameters	Additional Information
System.String	getAdditionalServicesRequest ()	
System.String	getAssetId ()	
System.String	getAssetTaskId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.DateTime	getProjectDeadline ()	
System.String	getProjectId ()	
System.String	getProjectName ()	

Return Type	Method and Parameters	Additional Information
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String	getTargetLanguageCode ()	
void	setAdditionalServicesRequest (System.String additionalServicesRequest)	
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	
void	setName (System.String name)	
void	setProjectDeadline (System.DateTime projectDeadline)	
void	setProjectId (System.String projectId)	
void	setProjectName (System.String projectName)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTags (System.String tags)	

Return Type	Method and Parameters	Additional Information
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

5.1.11 Receiving Updated Content to Translate (StartNeedAssetTaskUpdate)

Class	StartNeedAssetTaskUpdate
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	StartNeedAssetTaskUpdate ()
Supported Versions	Version 3.0 and higher

The Clay Tablet Platform launches the `StartNeedAssetTaskUpdate` event and sends it to the Translation Connector when there is updated content for the translation provider to translate.

For information about asset-task-update events, see page 35.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	
System.String	getAssetTaskId ()	

Return Type	Method and Parameters	Additional Information
System.String	getFileExt ()	
"FileType"	getFileType ()	
System.String	getProjectId ()	
System.String	getSourceLanguageCode ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (StartNeedAssetTaskUpdate eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm."FileType" fileType)	
void	setProjectId (System.String projectId)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

5.1.12 Receiving Content to Correct a Translation (StartNeedTranslationCorrectionAssetTask)

Class	StartNeedTranslationCorrectionAssetTask
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	StartNeedTranslationCorrectionAssetTask ()
Supported Versions	Version 3.0 and higher

The Clay Tablet Platform launches the `StartNeedTranslationCorrectionAssetTask` event and sends it to the Translation Connector when a translation correction is required.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetId ()	
System.String	getAssetTaskId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.DateTime	getProjectDeadline()	

Return Type	Method and Parameters	Additional Information
System.String	getProjectId()	
System.String	getProjectName ()	
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String	getTargetLanguageCode ()	
System.String	RanderToXML (StartNeedTranslationCorrectionAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm."FileType" fileType)	
void	setName (System.String name)	
void	setProjectDeadline (System.DateTime projectDeadline)	
void	setProjectId(System.String projectId)	
void	setProjectName (System.String projectName)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	

Return Type	Method and Parameters	Additional Information
void	setTags (System.String tags)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.13 Receiving New Supporting Content (StartSupportAsset)

Class	StartSupportAsset
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	StartSupportAsset ()
Supported Versions	All versions

The Clay Tablet Platform launches the `StartSupportAsset` event and sends it to the Translation Connector. It launches this event after the content producer sends supporting content (a *support asset*) to the translation into provider. This type of content is not translated.

For information about support-asset events, see page 27.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetId ()	

Return Type	Method and Parameters	Additional Information
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.String	getProjectId ()	
System.String	getSupportAssetId ()	
System.String	getSupportAssetNativeId ()	
System.String	getTags ()	
System.String	RanderToXML (StartSupportAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetId (System.String assetId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	
void	setName (System.String name)	
void	setProjectId (System.String projectId)	

Return Type	Method and Parameters	Additional Information
void	setSupportAssetId (System.String supportAssetId)	
void	setSupportAssetNativeId (System.String supportAssetNativeId)	
void	setTags (System.String tags)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.1.14 Receiving Corrected Translated Content for Updating the Translation Memory (StartUpdateTMAsset)

Class	StartUpdateTMAsset
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	StartUpdateTMAsset ()
Supported Versions	Version 3.0 and higher

The Clay Tablet Platform launches the `StartUpdateTMAsset` event and sends it to the Translation Connector. It launches this event when the content producer sends corrected translated content to the translation provider. This event instructs the translation provider to update the translation memory with this corrected translated content.

For information about update-TM-asset events, see page 38.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	
IJobMetadata	getJobMetadata ()	
System.String	getRelatedAssetTaskId ()	
System.String	getSourceLanguageCode ()	
System.String	getTargetLanguageCode ()	
System.String	getUpdateTMAssetId ()	
System.String	RanderToXML (StartUpdateTMAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	
void	setRelatedAssetTaskId (System.String relatedAssetTaskId)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTargetLanguageCode (System.String targetLanguageCode)	

Return Type	Method and Parameters	Additional Information
void	setUpdateTMAssetId (System.String updateTMAssetId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.2 Code Samples: Receiving Events

This section contains the following examples of the Translation Connector receiving events:

- "Code Sample: Receiving Events" on page 120
- "Code Sample: Handling Errors and Receiving Translated Files" on page 84

Note: All sample code is also included in the `Samples` folder in the SDK package.

5.2.1 Code Sample: Receiving Events

The following code sample illustrates how the Translation Connector uses the events described in this section to receive events.

Note: The following sample code is also included in the SDK package, in the `\Samples\ProviderSample_DotNet\ProviderSample\ProviderReceiver\MyProviderReceiver.cs` file.

```
public class MyProviderReceiver
{
  // The ConnectionContext configuration file
  private  ConnectionContext context;
  private ProviderAssetTaskTranslationMap assetTaskTranslationMap;

  private StorageClientService storageClientService;
private com.claytablet.service.Event.ProviderSender sender;

public MyProviderReceiver(ConnectionContext context,
            StorageClientService storageClientService, ProviderSender sender)
  {
  // Pass a ConnectionContext to retrieve the path where
  // the translated files will be saved.
  this.context = context;

  // Pass a StorageClientService to download a file for translation.
  this.storageClientService = storageClientService;

  /*ProviderJobMapping is a helper class that helps map the Clay Tablet
  * information to your TMS information.
  * ProviderAssetTaskTranslationMap is a helper class to handle
  * saving/deleting/searching the ProviderJobMapping objects.
  *
  * If supported, you can save the Clay Tablet information to your TMS system.
  * Then the ProviderJobMapping and ProviderAssetTaskTranslationMap classes
  * are not required.
  */
  this.assetTaskTranslationMap =
  ProviderAssetTaskTranslationMap.Instance;

  // Pass a ProducerSender to send out events and files, if needed.
  // Refer to the Sample-Sender folder for examples.
  this.sender = sender;

}

public HandleResult ReceiveEvent(IEvent cttEvent)
{

  if (cttEvent is ApprovedAssetTask)
    return receiveEvent((ApprovedAssetTask)cttEvent);
  else if (cttEvent is CanceledAssetTask)
    return receiveEvent((CanceledAssetTask)cttEvent);
  else if (cttEvent is CanceledSupportAsset)
    return receiveEvent((CanceledSupportAsset)cttEvent);
  else if (cttEvent is ProcessingError)
    return receiveEvent((ProcessingError)cttEvent);
  else if (cttEvent is RejectedAssetTask)
    return receiveEvent((RejectedAssetTask)cttEvent);
  else if (cttEvent is StartAssetTask)
    return receiveEvent((StartAssetTask)cttEvent);
  else if (cttEvent is StartSupportAsset)
    return receiveEvent((StartSupportAsset)cttEvent);
  else
  {
    HandleResult handleResult = new HandleResult();
    handleResult.Success = false;
    handleResult.CanDeleteMessage = true;
    handleResult.ErrorMessage = "Can't identify the Event Type.";
    return handleResult;
  }

}

private HandleResult receiveEvent(ApprovedAssetTask curEvent)
{

  HandleResult handleResult = new HandleResult();
  // Insert the provider integration code here.
  // Add your code here to handle the ApprovedAssetTask event.
  return handleResult;

}

private HandleResult receiveEvent(CanceledAssetTask curEvent)
{

  HandleResult handleResult = new HandleResult();
  try {
    // The information from JobMapping
    ProviderJobMapping mapJob = null;

    // Find the providerFileId information from assetTaskMap.
    foreach (ProviderJobMapping mapping in assetTaskTranslationMap.ListJobs() )
    {
    if ( mapping.getAssetTaskId().Equals(curEvent.getAssetTaskId()) )
      {
        //Found
        mapJob = mapping;
        break;
      }
    }

    // Call the TMS to cancel the job.
    if ( mapJob != null  )
    {

      String tmsProjectId = mapJob.getTmsProjectGUID();
      String tmsDocumentId = mapJob.getTmsDocumentGUID();

      String tms_Url = mapJob.getServerUrl(); ;
      String tms_LoginName = mapJob.getLoginName() ;
      String tms_LoginPassword = mapJob.getLoginPassword();

      // Connect to the TMS to cancel ....

      handleResult.CanDeleteMessage = true;
      handleResult.Success = true;

    }
  } catch (Exception e) {

    handleResult.CanDeleteMessage = false;
    handleResult.Success = false;
    handleResult.ErrorMessage = "[CanceledAssetTask] Error.";
    handleResult.Message = e.Message;
  }

return handleResult;
}

private HandleResult receiveEvent(CanceledSupportAsset curEvent)
{

  HandleResult handleResult = new HandleResult();
  // Insert the provider integration code here.
  // Put your code here to handle the CanceledSupportAsset event.
  return handleResult;

}

private HandleResult receiveEvent(ProcessingError curEvent)
{

  HandleResult handleResult = new HandleResult();
  // Insert the provider integration code here.
  // Put your code here to handle the ProcessingError event.
  return handleResult;

}

private HandleResult receiveEvent(RejectedAssetTask curEvent)
{

  HandleResult handleResult = new HandleResult();
  // Insert the provider integration code here.
  // Put your code here to handle the RejectedAssetTask event.
  return handleResult;

}

private HandleResult receiveEvent(StartAssetTask curEvent)
{

  HandleResult handleResult = new HandleResult();
  // call TMS to translate the AssetTask
  try {

    String tms_Url = null;
    String tms_LoginName = null;
    String tms_LoginPassword = null ;
    //TODO,  get connection info,  maybe from AppSettinng

    if (tms_Url != null && tms_LoginName != null && tms_LoginPassword != null )
    {
      try
        {

          // Download the latest asset task revision file to local tmp folder.
          String downloadFilePath = storageClientService
                   .downloadLatestAssetTaskVersion(curEvent.getAssetTaskId(),
                   context.getTargetDirectory());

          String tms_SoureLngLCID = null;
          String tms_TargetLngLCID = null;
          // You may need to convert the Clay Tablet language code
          // to the TMS language code.

          // Check whether this project already exists.
          String tmsProjectGUID = assetTaskTranslationMap
                   .SearchSameTmsProjectGUID(curEvent.getProjectId());

          if (tmsProjectGUID == null)
            {
            // If this project does not exist in the TMS,
            // create a TMS project.
            }

          Console.WriteLine("tmsProjectGUID:"+tmsProjectGUID);
```

```
String tmpdocumentGUID = null;
// Add a project document to the TMS.

// Save information to ProviderJobMapping
ProviderJobMapping jobMapping = new ProviderJobMapping()

jobMapping.setAssetTaskId(curEvent.getAssetTaskId());
jobMapping.setAssetId(curEvent.getAssetId());
jobMapping.setProjectId(curEvent.getProjectId());

jobMapping.setTmsDocumentGUID(tmpdocumentGUID);
jobMapping.setLocalFileGUID( IdGenerator.createId() );
jobMapping.setTmsProjectGUID(tmsProjectGUID);

jobMapping.setFileExt(curEvent.getFileExt());
jobMapping.setFileName(System.IO.Path.GetFileName(downloadFilePath));
jobMapping.setLoginName( tms_LoginPassword );
jobMapping.setLoginPassword( tms_LoginName );
jobMapping.setServerUrl( tms_Url );

jobMapping.setSourceCttLanguageCode(
         curEvent.getSourceLanguageCode() );
jobMapping.setSourceTmsLanguageCode( tms_SoureLngLCID );
jobMapping.setTargetCttLanguageCode(
         curEvent.getTargetLanguageCode());
jobMapping.setTargetTmsLanguageCode( tms_TargetLngLCID );

// Call helper class to handle saving information.
assetTaskTranslationMap.Add(jobMapping);

// Send out the AcceptAssetTask event to notify the Clay Tablet
// Platform.
AcceptAssetTask acceptEvent = new AcceptAssetTask();
acceptEvent.setAssetTaskId(curEvent.getAssetTaskId() );
acceptEvent.setAssetTaskNativeId( tmpdocumentGUID );
acceptEvent.setEventId( IdGenerator.createId() );
```

```
    // Call ProviderSender to send an event.
    sender.sendEvent( acceptEvent );

    // Event handling is finished, so the message can be removed from the
    // SQS queue.
    handleResult.CanDeleteMessage = true;
    handleResult.Success = true;

  }
  catch (Exception exp)  //should catch TMS exception here.
  {
    handleResult.CanDeleteMessage = false;
    handleResult.Success = false;
    handleResult.ErrorMessage = "[StartAssetTask] Error.";
    handleResult.Message = exp.Message;

  }
}
else
  {
    handleResult.CanDeleteMessage = false;
    handleResult.Success = false;
    handleResult.ErrorMessage = "[StartAssetTask] Error.";
    handleResult.Message = "Can't find server config in
              ConnectionContext.xml file.";
  }


}
catch (Exception e) //catch general exception here.
  {

    handleResult.CanDeleteMessage = false;
    handleResult.Success = false;
    handleResult.ErrorMessage = "[StartAssetTask] Error.";
    handleResult.Message = e.Message;
  }
```

```
          return handleResult;
        }

        private HandleResult receiveEvent(StartSupportAsset curEvent)
        {

          HandleResult handleResult = new HandleResult();
          //TODO - provider integration code goes here.
          //Put your code here to handle StartSupportAsset event
          return handleResult;

        }
      }
```

5.2.2 Code Sample: Retrieving Messages, Deserializing Event Objects, and Handling Events

The following code sample illustrates how the Translation Connector uses the events described in this section to retrieve messages, deserialize event objects, and handle events.

Note: The following sample code is also included in the SDK package, in the `\Samples\ProviderSample_DotNet\ProviderSample\ProviderReceiver\ReceiverTest.cs` file.

```
public class ReceiverTest
{
  static void Main(string[] args)
  {

    SourceAccountProvider sap;
    TargetAccountProvider tap;

    QueueSubscriberService queueSubscriberService;
    QueuePublisherService queuePublisherService;
    StorageClientService storageClientService;
    MyProviderReceiver receiver;

    ConnectionContext context;
    ProviderSender sender;
```

```
/*
* Sample configuration setting
*
* Check the app.config in the sample.
* For web applications, configuration is in the web.config file.
*
* When configuring the three keys below, please use an absolute path.
* Since Clay Tablet may be used in web applications, the absolute path
* can be configured outside the web folder, to keep account information
* and connectionContent information info more secure.
*
  <appSettings>
    <add key="CTT2_SourceAccount"
                value="E:\...\ProviderSample\accounts\source.xml" />
    <add key="CTT2_TargetAccount"
                value="E:\...\ProviderSample\accounts\target.xml" />
    <add key="CTT2_ConnectionContext_Folder"
                value="E:\...\ProviderSample\data\" />
  </appSettings>
*
*
*/

// Load ConnectionContext, so that the SDK looks for a file called
// "connectionContext.xml" in the folder specified in the "CTT2_
ConnectionContext_Folder" setting.

// Refer to the documentation for more information.
context = new ConnectionContext();
context.load();

//Initialize a source account so that theSDK looks for the source account
//file that appsetting ("CTT2_SourceAccount") specifies.
sap = new SourceAccountProvider();

// Initialize a target account so that the SDK looks for the source account
// file that appsetting ("CTT2_TargetAccount") specifies.
```

```
tap = new TargetAccountProvider();

Account sourceAccount = sap.get();

storageClientService = new StorageClientServiceS3();
storageClientService.setPublicKey(sourceAccount.getPublicKey());
storageClientService.setPrivateKey(sourceAccount.getPrivateKey());
storageClientService.setStorageBucket(sourceAccount.getStorageBucket());

queuePublisherService = new QueuePublisherServiceSQS();
queuePublisherService.setPublicKey(sourceAccount.getPublicKey());
queuePublisherService.setPrivateKey(sourceAccount.getPrivateKey());
queuePublisherService.setEndpoint(sourceAccount.getQueueEndpoint());

queueSubscriberService = new QueueSubscriberServiceSQS();
queueSubscriberService.setPublicKey(sourceAccount.getPublicKey());
queueSubscriberService.setPrivateKey(sourceAccount.getPrivateKey());
queueSubscriberService.setEndpoint(sourceAccount.getQueueEndpoint());

// Initialize a ProviderSender, because the receiver may need to send an
event back in response.
sender = new ProviderSender(context,sap,tap, queuePublisherService,
            storageClientService);

// Initialize your ProviderReceiver.
receiver = new MyProviderReceiver(context, storageClientService, sender);

Console.WriteLine(string.Format("Check SQS message for items needing
            translation. Start at {0}\t", System.DateTime.Now.ToString()) );

// Retrieve all avaiable messages.
int maxMessages = 0;

com.claytablet.queue.model.Message[] messages =
            queueSubscriberService.receiveMessages(maxMessages);

Console.WriteLine("Found " + messages.Length + " messages." );
```

```
if (messages.Length > 0)
{
  int msg_Count = 0;
  foreach (com.claytablet.queue.model.Message message in messages)
  {
    msg_Count++;
    Console.WriteLine(msg_Count.ToString() + ")Message Body:\n" +
                message.getBody() );

    try
    {
      // Deserializing, from XML to AbsEvent.
      IEvent curEvent = AbsEvent.fromXml(message.getBody());

      // Call your ProducerReceiver to handle the event.
      HandleResult curHandleResult = receiver.ReceiveEvent(curEvent);

      if (curHandleResult.CanDeleteMessage)
      {
        // After the event has been handled, delete it from the queue.
        queueSubscriberService.deleteMessage(message);
        Console.WriteLine("Message handled, so it can be
                deleted from queue." );
      }

      if (!curHandleResult.Success)
      {
        Console.WriteLine("Event handling error.\nError
                Message:" + curHandleResult.ErrorMessage );
      }

    }
    catch (Exception e)
    {
      Console.WriteLine(e.Message );
    }
```

```
        }
      }

      Console.WriteLine("Press any key to exit...");
      Console.ReadLine();

    }
  }
```

5.3 Retrieving Job Metadata

Reference	com.claytablet.model.Event.metadata
Supported Versions	Version 3.2 and higher

You retrieve job metadata so that you can identify all the content (*assets*) in a translation job.

1. In the following event messages that you receive, use the `getJobMetadata()` method to retrieve the IJobMetadata interface attached to these messages.

   Note: If there was no job metadata attached to the original messages sent by the content producer, the `getJobMetadata()` method returns null.

   - StartAssetTask
   - StartSupportAssetTask
   - StartAssetTaskQuote
   - ApprovedAssetTaskQuote
   - StartNeedTranslationCorrectionAssetTask
   - StartUpdateTMAsset

2. In the `IJobMetadata` interface, use the `getJobId()` method to determine which content (*assets*) and supporting content (*support assets*) belong to the same job.

3. In the `IJobMetadata` interface, use the `getAssetTaskCount()` method to determine whether you have received all required content (assets) for the job.

4. In the `IJobMetadata` interface, use the `getSupportAssetCount()` method to determine whether you have received all required support assets for the job.

5.4 Events a Translation Connector Can Send

Reference	com.claytablet.model.Event.provider

A Translation Connector can launch the following events:

Event	Description
AcceptAssetTask	Accepts translating the content into the specified target language.
AcceptNeedAssetTaskUpdate	Accepts updating the translation for the content it previously translated.
AcceptNeedTranslationCorrectionAssetTask	Accepts correcting the translated content it previously translated.
AcceptUpdateTMAsset	Accepts updating the translation memory with the updated content.
AssetTaskProviderError	Notifies the Clay Tablet Platform and the content producer about any issue encountered during translation.
SubmitAssetTask	Submits the translated content to the content producer for review.
SubmitTranslationCorrectionAssetTask	Submits the corrected translated content to the content producer for review.
SubmitUpdatedAssetTask	Submits the updated translated content to the content producer for review.
UpdateAssetTaskState	Sends an update about the state and/or the estimated arrival time of content translated to a particular language (an *asset task*) to the content producer.

5.4.1 Accepting a Translation Task (AcceptAssetTask)

Class	AcceptAssetTask
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent

Constructor	AcceptAssetTask()
Supported Versions	All versions

The Translation Connector launches the `AcceptAssetTask` event after receiving the StartAssetTask event from the Clay Tablet Platform, which is a request from the content producer to translate content into specified target languages. (Each target language represents a separate *asset task*.) The translation provider uses this event to notify the Clay Tablet Platform and the content producer that it accepts translating the content into the specified target language.

Note: This event includes the `getAssetTaskNativeId` method. The native ID field is the translation provider's identifier for the asset task.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getAssetTaskNativeId ()	
System.String	RanderToXML (AcceptAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setAssetTaskNativeId (System.String assetTaskNativeId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.2 Accepting Updated Content to Translate (AcceptNeedAssetTaskUpdate)

Class	AcceptNeedAssetTaskUpdate
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	AcceptNeedAssetTaskUpdate ()
Supported Versions	Version 3.0 and higher

The Translation Connector launches the `AcceptNeedAssetTaskUpdate` event after receiving the StartNeedAssetTaskUpdate event from the Clay Tablet Platform, which is a request from the content producer to update the translation for the content it previously translated. The translation provider uses this event to notify the Clay Tablet Platform and the content producer that it accepts updating this translated content.

For information about asset-task-update events, see page 35.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetTaskId ()	
System.String	RanderToXML (AcceptNeedAssetTaskUpdate eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.3 Accepting Corrections to Translated Content (AcceptNeedTranslationCorrectionAssetTask)

Class	AcceptNeedTranslationCorrectionAssetTask
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	AcceptNeedTranslationCorrectionAssetTask ()
Supported Versions	Version 3.0 and higher

The Translation Connector launches the `AcceptNeedTranslationCorrectionAssetTask` event after receiving the StartNeedTranslationCorrectionAssetTask event from the Clay Tablet Platform, which is a request from the content producer to correct the translation for the content it previously translated. The translation provider uses this event to notify the Clay Tablet Platform and the content producer that it accepts correcting this translated content.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetTaskId ()	
System.String	getAssetTaskNativeId ()	
System.String	RanderToXML (AcceptNeedTranslationCorrectionAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setAssetTaskNativeId (System.String assetTaskNativeId)	

Return Type	Method and Parameters	Additional Information
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.4 Accepting Corrected Translated Content for Updating the Translation Memory (AcceptUpdateTMAsset)

Class	AcceptUpdateTMAsset
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	AcceptUpdateTMAsset ()
Supported Versions	Version 3.0 and higher

The Translation Connector launches the `AcceptUpdateTMAsset` event after receiving the StartUpdateTMAsset event from the Clay Tablet Platform, which is a request from the content producer to update the translation memory with the corrected translated content it provided. The translation provider uses this event to notify the Clay Tablet Platform and the content producer that it accepts updating the translation memory with the updated content.

For information about update-TM-asset events, see page 38.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getRelatedAssetTaskId ()	

Return Type	Method and Parameters	Additional Information
System.String	getUpdateTMAssetId ()	
System.String	RanderToXML (AcceptUpdateTMAsset eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setRelatedAssetTaskId (System.String relatedAssetTaskId)	
void	setUpdateTMAssetId (System.String updateTMAssetId)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.5 Notifying the Producer of an Error (AssetTaskProviderError)

Class	AssetTaskProviderError
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructors	- AssetTaskProviderError () - AssetTaskProviderError (System.String assetTaskId, System.String subject, System.String errorMessage, System.String errorDetails, boolean fatal)
Supported Versions	Version 3.3 and higher

When a translation provider runs into issues during translation (processing an asset task), it should send an `AssetTaskProviderError` event to notify the Platform and the content producer about the issue.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getErrorDetails ()	
System.String	getErrorMessage ()	
System.String	getSubject ()	See the note below for additional information.
boolean	isFatal ()	See the note below for additional information.
void	setAssetTaskId (System.String assetTaskId)	
void	setErrorDetails (System.String errorDetails)	
void	setErrorMessage (System.String errorMessage)	
void	setFatal (System.boolean fatal)	See the note below for additional information.
void	setSubject (System.String subject)	See the note below for additional information.

Note about `isFatal` and `setFatal`: During a *fatal* error, the provider has permanently stopped processing the asset task, and human intervention is required to process the asset task. During a *non-fatal* error, the provider retries processing the same asset task: if the underlying reason for the error is resolved, the the provider can successfully process the asset task without intervention.

Note about `getSubject` and `setSubject`: Sometimes, when there is a transient error, the provider retries processing an asset task. This may cause the same processing error to occur repeatedly. The subject field enables the platform and producer to detect repetitive errors. If the provider sends two `AssetTaskProviderError` events for the same asset task with same subjects, without any other event in between, then the second is a repeat of the first.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.6 Submitting Translated Content for Review (SubmitAssetTask)

Class	SubmitAssetTask
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	SubmitAssetTask ()
Supported Versions	All versions

The Translation Connector launches the `SubmitAssetTask` event after the translation provider completes the content translation to a particular target language (an *asset task*). It uses this event to submit the translated content to the content producer for review.

For information about asset-task events, see page 28.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getNativeState ()	
System.String	RanderToXML (SubmitAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	

Return Type	Method and Parameters	Additional Information
void	setNativeState (System.String nativeState)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.7 Submitting the Corrected Translation for Review (SubmitTranslationCorrectionAssetTask)

Class	SubmitTranslationCorrectionAssetTask
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	SubmitTranslationCorrectionAssetTask ()
Supported Versions	Version 3.0 and higher

The Translation Connector launches the `SubmitTranslationCorrectionAssetTask` event after the translation provider corrects the content translation. It uses this event to submit the corrected translated content to the content producer for review.

For information about translation-correction-asset events, see page 37.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	getNativeState ()	
System.String	RanderToXML (SubmitTranslationCorrectionAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
void	setNativeState (System.String nativeState)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.8 Submitting Updated Translated Content for Review (SubmitUpdatedAssetTask)

Class	SubmitUpdatedAssetTask
Reference	com.claytablet.model.Event.provider
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	SubmitUpdatedAssetTask ()

Supported Versions	Version 3.0 and higher

The Translation Connector launches the `SubmitUpdatedAssetTask` event after the translation provider updates the translated content. It uses this event to submit the updated translated content to the content producer for review.

For information about updated-asset-task events, see page 36.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.String	getFileExt ()	
System.String	RanderToXML (SubmitUpdatedAssetTask eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setFileExt (System.String fileExt)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

5.4.9 Updating the Translation Status (UpdateAssetTaskState)

Class	UpdateAssetTaskState
Reference	com.claytablet.model.Event.provider

Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructor	UpdatedAssetTaskState ()
Supported Versions	All versions

The Translation Connector launches the `UpdateAssetTaskState` event to send an update about the state and/or the estimated arrival time of content translated to a particular language (an *asset task*) to the content producer.

For information about asset-task-state events, see page 33.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAssetTaskId ()	
System.DateTime	getEta ()	
System.String	getNativeState ()	
System.String	getNotes ()	
int	getTranslationPercentage ()	
System.String	RanderToXML (UpdateAssetTaskState eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.
void	setAssetTaskId (System.String assetTaskId)	
void	setEta (System.DateTime eta)	
void	setNativeState (System.String nativeState)	
void	setNotes (System.String notes)	
void	setTranslationPercentage (int translationPercentage)	

Return Type	Method and Parameters	Additional Information
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

5.4.10 Code Sample: Sending Translation Status or Translated Files

The following code sample illustrates how the Translation Connector uses the events described in this section to determine translation status, send a status update, or send the translated files, if translation is complete.

This code enables the Producer to get translation requests, download files that need translation, and send translated content back to the producer.

Note: The following sample code is also included in the SDK package, in the `\Samples\ProviderSample_DotNet\ProviderSample\ProviderSender\ProviderStatePollerTest.cs` file.

```
public class ProviderStatePollerTest
{
  static void Main(string[] args)
  {
    SourceAccountProvider sap;
    TargetAccountProvider tap;

    QueueSubscriberService queueSubscriberService;
    QueuePublisherService queuePublisherService;
    StorageClientService storageClientService;

    ConnectionContext context;
    ProviderSender sender;

    /*
```

```
* Sample configuration setting
*
* Check the app.config in the sample.
* For web applications, configuration is in the web.config file.
*
* When configuring the three keys below, please use an absolute path.
* Since Clay Tablet may be used in web applications, the absolute path
* can be configured outside the web folder, to keep account information
* and connectionContent information info more secure.
*
  <appSettings>
    <add key="CTT2_SourceAccount"
                 value="E:\...\ProviderSample\accounts\source.xml" />
    <add key="CTT2_TargetAccount"
                 value="E:\...\ProviderSample\accounts\target.xml" />
    <add key="CTT2_ConnectionContext_Folder"
                 value="E:\..\ProviderSample\data\" />
  </appSettings>
*
*
*/

// Load ConnectionContext so that the SDK looks for the
// "connectionContext.xml" file in the folder specified by the
// "CTT2_ConnectionContext_Folder" setting.
context = new ConnectionContext();
context.load();

// Initialize a source account, so that the SDK looks for the source account
/// file, specified by the "CTT2_SourceAccount" appsetting.
sap = new SourceAccountProvider();

// Initialize a target account, so that the SDK looks for the target account
// file, specified by the "CTT2_SourceAccount" appsetting.
tap = new TargetAccountProvider();

Account sourceAccount = sap.get();
```

```
storageClientService = new StorageClientServiceS3();
storageClientService.setPublicKey(sourceAccount.getPublicKey());
storageClientService.setPrivateKey(sourceAccount.getPrivateKey());
storageClientService.setStorageBucket(sourceAccount.getStorageBucket());

queuePublisherService = new QueuePublisherServiceSQS();
queuePublisherService.setPublicKey(sourceAccount.getPublicKey());
queuePublisherService.setPrivateKey(sourceAccount.getPrivateKey());
queuePublisherService.setEndpoint(sourceAccount.getQueueEndpoint());

queueSubscriberService = new QueueSubscriberServiceSQS();
queueSubscriberService.setPublicKey(sourceAccount.getPublicKey());
queueSubscriberService.setPrivateKey(sourceAccount.getPrivateKey());
queueSubscriberService.setEndpoint(sourceAccount.getQueueEndpoint());

//Initialize a ProviderSender, in case it needs to send an event back.
sender = new ProviderSender(context, sap, tap,
queuePublisherService, storageClientService);

ProviderAssetTaskTranslationMap assetTaskTranslationMap =
            ProviderAssetTaskTranslationMap.Instance;

Console.WriteLine("Polling TMS for asset task state changes.");

foreach (ProviderJobMapping mapping in assetTaskTranslationMap.ListJobs())
{

  String sourceLanguageCode = mapping.getSourceCttLanguageCode();
  String targetLanguageCode = mapping.getTargetCttLanguageCode();

  String tmsSourceLCID = mapping.getSourceTmsLanguageCode();
  String tmsTargetLCID = mapping.getTargetTmsLanguageCode() ;

  String tmsProjectGUID = mapping.getTmsProjectGUID();
  String tmsDocumentGUID = mapping.getTmsDocumentGUID();
  String localFileGUID = mapping.getLocalFileGUID();
  String tms_Url = mapping.getServerUrl();
  String tms_LoginName = mapping.getLoginName() ;
  String tms_LoginPassword = mapping.getLoginPassword() ;

  if ( tmsDocumentGUID != null && tmsProjectGUID != null )
  {
    Console.WriteLine("Check document Status:" + tmsDocumentGUID);
    try
    {

       Boolean  translationFinished = false;
       // Insert your code to check the translation status here.

       if ( translationFinished ) {
       // If a file translation is done, send it back with
       // submitAssetTask events.
       String downloadFileFolder = context.getTargetDirectory();
       String downloadFilePath = null;
       // Download the translated file from TMS to the local folder,
       // "downloadFileFolder".
       // The full file path goes to downloadFilePath.
       // Send out the event, notifying the Clay Tablet Platform that
       // a file translation is completed.
       SubmitAssetTask submitAssetTask = new SubmitAssetTask();
       submitAssetTask.setEventId( IdGenerator.createId() );

       // Set the original Clay Tablet AssetTaskId.
       submitAssetTask.setAssetTaskId(mapping.getAssetTaskId());
       submitAssetTask.setNativeState("Completed");
       submitAssetTask.setFileExt(downloadFilePath
            .Substring(downloadFilePath.LastIndexOf(".")+ 1,
                   downloadFilePath.Length ) );
       sender.sendEvent(submitAssetTask, downloadFilePath );
       Console.WriteLine("Send out translated file: " + downloadFilePath);

       // The job is done, so remove this job from Mapping.
```

```
assetTaskTranslationMap.Remove(mapping);

// If needed, you can add code to delete the job or delete the
// project file from the TMS.
}
else
{

  // Check the  translation status. If the status changed,
  // send an UpdateAssetTaskState event.
  String lastStatus = mapping.getTmsTranslationStatus();
  String tmsCurrentStatus = null;

  // Retrieve the translation status from the TMS, and
  // put into"tmsCurrentStatus".
  if (tmsCurrentStatus != null && lastStatus != null )
  if (! tmsCurrentStatus.Equals(lastStatus))
    {
    // Status Changed
    UpdateAssetTaskState updateAssetTaskState = new
    UpdateAssetTaskState();
    updateAssetTaskState.setAssetTaskId(mapping.getAssetTaskId());
    updateAssetTaskState.setEventId(IdGenerator.createId());
    updateAssetTaskState.setNativeState(tmsCurrentStatus);
    /* Set the percentage of translation done.
    * Since Clay Tablet can not map all the native states from
    * every TMS system to a state in Clay Tablet, the
    * Translation Percentage provides very useful information
    * to Producer (CMS) side. Even the native state from the TMS
    * is not very user friendly.
    *
    * To fire an UpdateAssetTaskState event, the Provider must
    * set up the TranslationPercentage to avoid the default of 0
    * percent being sent back to the Producer.
    event.setTranslationPercentage(50);

    */
```

```
            sender.sendEvent(updateAssetTaskState);
          }

        }

      }
      catch (Exception e) //Catch a TMS exception here.
        {
          //Add relevant code.
        }
      }

    }

    Console.WriteLine("Press any key to exit...");
    Console.ReadLine();
  }
}
```

6 Events that All Connectors Receive

Both a Content Connector (for a content producer) and a Translation Connector (for a translation provider) can receive the following events (classes):

- ProcessingError
- AssetTaskProcessingError

6.1 Viewing a Processing Error (ProcessingError)

Class	ProcessingError
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Implements	IEvent
Constructors	ProcessingError () ProcessingError (System.String errorMessage, System.String errorDetails)
Supported Versions	All versions

If an error occurs when the Clay Tablet Platform is processing an event that has been sent to it, the Clay Tablet Platform launches the `ProcessingError` event. It sends this event to both the Content Connector and the Translation Connector.

Tip for the Content Producer: Examine the processing error, because it usually indicates necessary action by the producer. For example, an error may occur if an asset was created, but the Platform could not process it because the file had not been uploaded.

For information about processing-error events, see page 38.

This class has the following methods:

Return Type	**Method and Parameters**	**Additional Information**
System.String	getErrorDetails ()	
System.String	getErrorMessage ()	
System.String	RanderToXML (ProcessingError eventObject)	This method displays the message in XML format. It is useful when the logs include these messages.

Return Type	Method and Parameters	Additional Information
void	setErrorDetails (System.String errorDetails)	
void	setErrorMessage (System.String errorMessage)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ( )	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals	getEventId	getXStream	setEventId	toString
fromXml	getSourceAccountId	hashCode	setSourceAccountId	toXml
getCreateDate	getTargetAccountId	setCreateDate	setTargetAccountId	

6.1.1 Viewing a Translation Processing Error (AssetTaskProcessingError)

Class	AssetTaskProcessingError
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.platform.ProcessingError
Implements	IEvent
Constructor	AssetTaskProcessingError ()
Supported Versions	Version 3.3 and higher

The `AssetTaskProcessingError` event is a subclass of the `ProcessingError` event, and it extends this class. It describes an error that occurs specifically to an asset task, with additional information to identify the asset task involved, and information about whether additional intervention is required.

For information about processing-error events, see page 38.

This class has the following methods:

Return Type	Method and Paramaters	Additional Information
System.String	getAssetId ()	
System.String	getAssetTaskId ()	
System.String	getProjectId ()	
int	getRepeat ()	*Repeat* indicates to the Content Connector whether the error is a repetition of a previous error, and how many times the error has been repeated. If the value is `0`, the error is not a repeat. Generally the Clay Tablet Platform prevents sending repeat errors to the producer.
System.String	getSubject ()	Sometimes the Clay Tablet Platform or the Translation Connector can retry processing an asset task in case of transient error, which may resulting in the same processing error occurs repeatedly. The purpose of the subject field is to enable the Platform and the Content Connector to detect repetitive errors. If two `AssetTaskProcessingError` event messages for the same asset task with same subjects arrive, without any other event in between, then the second is a repeat of the first.
bool	isFatal ()	A *fatal* error means that the processing of the asset task has stopped (either at the Platform or at the Provider), and human intervention is required to process the asset task. A *non-fatal* error means that the Platform or Provider will retry processing the asset task: if the underlying reason for the error is resolved, the asset task can be processed successfully without intervention.
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setFatal (System.boolean fatal)	A *fatal* error means that the processing of the asset task has stopped (either at the Platform or at the Provider), and human intervention is required to process the asset task. A *non-fatal* error means that the Platform or Provider will retry processing the asset task: if the underlying reason for the error is resolved, the asset task can be processed successfully without intervention.

Return Type	Method and Paramaters	Additional Information
void	setProjectId (System.String projectId)	
void	setRepeat (int repeat)	*Repeat* indicates to the Content Connector whether the error is a repetition of a previous error, and how many times the error has been repeated. If the value is `0`, the error is not a repeat. Generally the Clay Tablet Platform prevents sending repeat errors to the producer.
void	setSubject (System.String subject)	Sometimes the Clay Tablet Platform or the Translation Connector can retry processing an asset task in case of transient error, which may resulting in the same processing error occurs repeatedly. The purpose of the subject field is to enable the Platform and the Content Connector to detect repetitive errors. If two `AssetTaskProcessingError` event messages for the same asset task with same subjects arrive, without any other event in between, then the second is a repeat of the first.

This class inherits the following methods from the parent com.claytablet.model.Event.platform.ProcessingError class:

- getErrorDetails
- getErrorMessage
- setErrorDetails
- setErrorMessage
- validate

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

7 Job-Related Interfaces and Metadata-Related Classes

The following interfaces group content into jobs, and provide job-related metadata. For more information, see "Grouping Assets into Jobs Using Job Metadata Interfaces" on page 41.

Job-Related Interface	Description
IJobMetadata	Defines a basic set of common metadata for a job, as well as an extensible set of additional translation-provider-specific metadata.
IAttribute	Specifies the names and values of metadata of assets in a job. This extends the standard set of job metadata with translation-provider-specific metadata.
IMetadata	Specifies the metadata for an asset in a job. This extends the standard set of job metadata with translation-provider-specific metadata.
IMetadataGroup	Specifies the metadata for a group of assets in a job.

The following classes provide metadata for jobs:

Metadata-Related Class	Description
Metadata	Captures additional information, which is useful when creating a custom integration from the producer to the provider.
JobMetadata	Contains methods for setting the metadata for a project.
MetadataFactory	Contains a group of static factory methods to create instances of job metadata-related interfaces. You use these factory methods instead of directly instantiating implementation classes of these interfaces.
MetadataException	Provides data about a metadata exception.

7.1 IJobMetadata Interface

Interface	IJobMetadata
Reference	com.claytablet.model.Event.metadata
Known Implementing Classes	JobMetadata
Supported Versions	All versions

This interface defines a basic set of common metadata for a job, as well as an extensible set of additional translation-provider-specific metadata.

This interface has the following methods:

Return Type	Method and Parameters	Additional Information
IJobMetadata	addMetadataGroup (IMetadataGroup metadataGroup)	This method associates and retrieves additional translation-provider-specific metadata sets for a job.
List< IMetadataGroup >	getAdditionalMetadataGroups ()	This method associates and retrieves additional translation-provider-specific metadata sets for a job.
int	getAssetTaskCount ()	The Translation Connector uses a job's asset task count to determine whether it has received all the asset tasks in a job. Each target language for an asset is a separate asset task. For example, suppose a job contains three assets: two assets have three target languages, and one asset has four target languages. This job has ten asset tasks. Before you can send a `CreateAsset` message, you must calculate the total asset task count for the entire job. Support assets are not included in the asset task count.
System.String	getClient ()	
System.String	getCreator ()	
System.String	getCreatorContact ()	
System.String	getDescription ()	
System.String	getDueDate ()	
System.String	getJobId ()	Each job must have a unique job ID (typically a GUID), which the Translation Connector uses to determine the job association of assets tasks. If you are sure that your project contains only one job, this ID can be the project ID. However, if your project can contain multiple jobs, it should be a different ID.
System.String	getJobName ()	

Return Type	Method and Parameters	Additional Information
IMetadataGroup	getMetadataGroup (IMetadataGroup outputGroup)	This method associates and retrieves additional translation-provider-specific metadata sets for a job. For more information, see "IMetadataGroup Interface" on page 162.
System.String	getPoReference ()	
System.String	getProducerId ()	
System.String	getStartDate ()	
int	getSupportAssetCount ()	The Translation Connector uses a job's support asset count to determine whether it has received all the support assets in a job.
IJobMetadata	setAssetTaskCount (int assetTaskCount)	The Translation Connector uses a job's asset task count to determine whether it has received all the asset tasks in a job. Each target language for an asset is a separate asset task. For example, suppose a job contains three assets: two assets have three target languages, and one asset has four target languages. This job has ten asset tasks.Before you can send a `CreateAsset` message, you must calculate the total asset task count for the entire job. Support assets are not included in the asset task count.
IJobMetadata	setClient (System.String client)	
IJobMetadata	setCreator (System.String creator)	
IJobMetadata	setCreatorContact (System.String creatorContact)	
IJobMetadata	setDescription (System.String description)	
IJobMetadata	setDueDate (System.String dueDate)	

Return Type	Method and Parameters	Additional Information
IJobMetadata	setJobId (System.String jobId)	Each job must have a unique job ID (typically a GUID), which the Translation Connector uses to determine the job association of assets tasks. If you are sure that your project contains only one job, this ID can be the project ID. However, if your project can contain multiple jobs, it should be a different ID.
IJobMetadata	setJobName (System.String jobName)	
IJobMetadata	setPoReference (System.String poReference)	
IJobMetadata	setProducerId (System.String producerId)	
IJobMetadata	setStartDate (System.String startDate)	
IJobMetadata	setSupportAssetCount (int supportAssetCount)	The Translation Connector uses a job's support asset count to determine whether it has received all the support assets in a job.
System.String	toXml ()	

7.1.1 JobMetadata Class

Class	JobMetadata
Reference	com.claytablet.model.Event.metadata
Implements	IJobMetadata
Constructor	JobMetadata ()
Supported Versions	All versions

This class contains methods for setting the metadata for a project.

The following example illustrates how to create it:

```
IJobMetadata jobMetadata = MetadataFactory.newJobMetadata();
jobMetadata.setJobId(projectId);
jobMetadata.setJobName("my translation job");
jobMetadata.setPoReference("poreference");
```

The `JobMetadata` class has the following methods:

Return Type	Method and Parameters	Additional Information
IJobMetadata	addMetadataGroup (IMetadataGroup metadataGroup)	Specified by `addMetadataGroup` in the `IJobMetadata` interface.
JobMetadata	FromXML (System.String xmlString)	Throws MetadataException.
List< IMetadataGroup >	getAdditionalMetadataGroups ()	Specified by `getAdditionalMetadataGroups` in the `IJobMetadata` interface.
int	getAssetTaskCount ()	Specified by `getAssetTaskCount` in the `IJobMetadata` interface.
System.String	getClient ()	Specified by `getClient` in the `IJobMetadata` interface.
System.String	getCreator ()	
System.String	getCreatorContact ()	Specified by `getCreatorContact` in the `IJobMetadata` interface.
System.String	getDescription ()	Specified by `getDescription` in the `IJobMetadata` interface.
System.String	getDueDate ()	Specified by `getDueDate` in the `IJobMetadata` interface.
System.String	getJobId ()	Specified by `getJobId` in the `IJobMetadata` interface.
System.String	getJobName ()	Specified by `getJobName` in the `IJobMetadata` interface.
IMetadataGroup	getMetadataGroup (IMetadataGroup outputGroup)	Specified by `getMetadataGroup` in the `IJobMetadata` interface.
System.String	getPoReference ()	Specified by `getPoReference` in the `IJobMetadata` interface.
System.String	getProducerId ()	Specified by `getProducerId` in the `IJobMetadata` interface.

Return Type	Method and Parameters	Additional Information
System.String	getStartDate ()	Specified by `getStartDate` in the `IJobMetadata` interface.
int	getSupportAssetCount ()	Specified by `getSupportAssetCount` in the `IJobMetadata` interface.
void	setAdditionalMetadataGroups (List<IMetadataGroup> additionalMetadataGroupList)	Specified by `setAdditionalMetadataGroups` in the `IJobMetadata` interface.
IJobMetadata	setAssetTaskCount (int assetTaskCount)	Specified by `setAssetTaskCount` in the `IJobMetadata` interface.
IJobMetadata	setClient (System.String client)	Specified by `setClient` in the `IJobMetadata` interface.
IJobMetadata	setCreator (System.String creator)	
IJobMetadata	setCreatorContact (System.String creatorContact)	Specified by `setCreatorContact` in the `IJobMetadata` interface.
IJobMetadata	setDescription (System.String description)	Specified by `setDescription` in the `IJobMetadata` interface.
IJobMetadata	setDueDate (System.String dueDate)	Specified by `setDueDate` in the `IJobMetadata` interface.
IJobMetadata	setJobId (System.String jobId)	Specified by `setJobId` in the `IJobMetadata` interface.
IJobMetadata	setJobName (System.String jobName)	Specified by `setJobName` in the `IJobMetadata` interface.
IJobMetadata	setPoReference (System.String poReference)	Specified by `setPoReference` in the `IJobMetadata` interface.
IJobMetadata	setProducerId (System.String producerId)	Specified by `setProducerId` in the `IJobMetadata` interface.
IJobMetadata	setStartDate (System.String startDate)	Specified by `setStartDate` in the `IJobMetadata` interface.
IJobMetadata	setSupportAssetCount (int supportAssetCount)	Specified by `setSupportAssetCount` in the `IJobMetadata` interface.

Return Type	Method and Parameters	Additional Information
System.String	toXml ()	Specified by `toXml` in the `IJobMetadata` interface.
System.String	toXML (JobMetadata jobMetadata)	Throws MetadataException.

7.2 IAttribute Interface

Interface	IAttribute
Reference	com.claytablet.model.Event.metadata
Known Implementing Class	Attr
Supported Versions	All versions.

The `IAttribute` interface specifies the names and values of metadata of assets in a job. This extends the standard set of job metadata with translation-provider-specific metadata.

This interface has the following methods:

Return Type	Method and Parameters
System.String	getName ()
System.String	getValue ()
void	setName (System.String name)
void	setValue (System.String value)

7.3 IMetadata Interface

Interface	IMetadata
Reference	com.claytablet.model.Event.metadata
Known Subinterfaces	IMetadataGroup
All Known Implementing Classes	BaseMetadataGroup, FreewayMetadataGroup, Metadata

The `IMetadata` interface specifies the metadata for an asset in a job. This extends the standard set of job metadata with translation-provider-specific metadata.

This interface has the following methods:

Return Type	Method and Parameters
void	addAttribute (IAttribute attribute)
void	addAttribute (System.String attribute_name, System.String attribute_value)
List< IAttribute>	getAttributes ()
System.String	getMetadataName ()
System.String	getMetadataValue ()
void	setMetadataName (System.String metadata_name)
void	setMetadataValue (System.String metadata_value)

7.4 IMetadataGroup Interface

Interface	IMetadataGroup
Reference	com.claytablet.model.Event.metadata
Extends/Superinterfaces	IMetadata
All Known Implementing Classes	BaseMetadataGroup, FreewayMetadataGroup

The IMetadataGroup interface specifies the metadata for a group of assets in a job.

Tip: Developers of the Translation Connector can create a new implementation of the IMetadataGroup interface that contains metadata specific to the translation provider.

This interface has the following methods:

Return Type	Method and Parameters
void	addAttributeToGroup (IAttribute attribute)
void	addMetadataToGroup (IMetadata metadata)
List< IMetadata>	getAdditionalMetadataList ()
List< IAttribute>	getGroupAttributes ()
System.String	getGroupName ()

Return Type	Method and Parameters
boolean	isTransferableFrom (IMetadataGroup group)
void	setGroupName (System.String groupName)
void	transferFrom (IMetadataGroup sourceGroup)

7.4.1 Metadata Class

Class	Metadata
Reference	com.claytablet.model.Event.metadata.impl.Metadata
Direct Known Subclass	BaseMetadataGroup
Implements	IMetadata
Constructor	■ Metadata () ■ Metadata (IMetadata metadata)
Supported Versions	All versions

This class is useful when creating a custom integration from the producer to the provider, where you want to capture additional information.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
void	addAttribute (IAttribute attribute)	Specified by `addAttribute` in the `IMetadata` interface.
void	addAttribute (System.String attribute_name, System.String attribute_value)	Specified by `addAttribute` in the `IMetadata` interface.
List< IAttribute>	getAttributes ()	Specified by `getAttributes` in the `IMetadata` interface.
System.String	getMetadataName ()	Specified by `getMetadataName` in the `IMetadata` interface.
System.String	getMetadataValue ()	Specified by `getMetadataValue` in the `IMetadata` interface.
void	setAttributes (List<IAttribute> attributes)	

Return Type	Method and Parameters	Additional Information
void	setMetadataName (System.String metadata_ name)	Specified by `setMetadataName` in the `IMetadata` interface.
void	setMetadataValue (System.String metadata_ value)	Specified by `setMetadataValue` in the `IMetadata` interface.

7.4.2 MetadataFactory Class

Class	MetadataFactory
Reference	com.claytablet.model.Event.metadata
Constructor	MetadataFactory ()
Supported Versions	All versions

The `MetadataFactory` class contains a group of static factory methods to create instances of job metadata-related interfaces. You use these factory methods instead of directly instantiating implementation classes of these interfaces.

Return Type	Method and Parameters	Additional Information
static IJobMetadata	getJobMetadataFromXml (System.String xml)	Throws MetadataException.
static IAttribute	newAttribute (System.String name, System.String value)	
static IJobMetadata	newJobMetadata ()	
static IMetadata	newMetadata ()	

7.4.2.1 MetadataException Class

Class	MetadataException
Reference	com.claytablet.model.Event.metadata
Extends	System.Exception

Constructors	■ MetadataException (System.String message) ■ MetadataException (System.String message, System.String jobId, System.String jobName, System.String producerId) ■ MetadataException (System.String message, System.String jobId, System.String jobName, System.String producerId, System.Exception cause) ■ MetadataException (System.String message, System.Exception cause)
Supported Versions	All versions

The `MetadataException` class provides data about a metadata exception. It has the following methods:

Return Type	**Method and Parameters**
System.String	getJobId ()
System.String	getJobName ()
System.String	getProducerId ()

8 Testing Your Connector

After you use this SDK to build your Connector, you can use two Clay Tablet Technologies demo files to test it.

There is one demo file to test a Content Connector, and there is another demo file to test a Translation Connector. These files are part of the SDK for .NET package. Both demo files are built with Microsoft Visual Studio 2005.

1. Unzip the relevant package (`..\CTT DotNET SDK\Samples\*Sample_DotNet.zip`) to a local folder, and then open them with Microsoft Visual Studio 2005.

2. Contact Clay Tablet Technologies to create a test account so that you can run the demo projects. For details, see "How to Contact Clay Tablet Support" on page 12.

3. Copy the `source.xml` and `target.xml` files, which Clay Tablet Technologies also provides, to a folder.

4. In your configuration file (`app.config` or `web.config`), specify the account file locations. For example:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<appSettings>
<add key="CTT2_SourceAccount"
value="E:\...\ProducerSample\accounts\source.xml" />
<add key="CTT2_TargetAccount"
value="E:\...\ProducerSample\accounts\target.xml" />
<add key="CTT2_ConnectionContext_Folder"
value="E:\...\ProducerSample\data\" />
</appSettings>
</configuration>
```

9 Appendix: High-Level .NET Reference Information

This chapter contains reference information about the following .NET entities:

.NET Entity	Description
"References" on page 167	Describes the sub-sections of the com.claytablet.model.Event part of the SDK you use to build a Connector, including which types of events (classes) are in which sub-section.
"Abstract Classes" on page 168	Abstract classes that extend classes (events). The abstract classes cannot be instantiated.
"Language Class" on page 177	The Language class includes the constants that represent the supported languages and language codes and the static methods for working with those language codes.
"IEvent Interface" on page 174	The IEvent Interface has methods for all classes that are events.
"Return Types" on page 191	Custom return types in this SDK.
■ "ConnectionContext Class" on page 187 ■ "StorageDirectoryProvider Interface" on page 188	A class and the related interface for retrieving and setting the connection context fields, such as the local directory paths.

9.1 References

You use the com.claytablet.model.Event part of the SDK to build a Connector. This has the following subsections:

Reference	Description
com.claytablet.model.Event.platform	This contains events (classes) that the Clay Tablet Platform sends.
com.claytablet.model.Event.producer	This contains events (classes) that the Content Connector (Producer) sends.
com.claytablet.model.Event.provider	This contains events (classes) that the Translation Connector (Provider) sends.
com.claytablet.model.Event.metadata	This contains interfaces and static factory methods for grouping multiple assets as a single job, and for providing metadata about jobs.

9.2 Abstract Classes

Classes (events) in the SDK reference the following abstract classes, which cannot be instantiated:

Abstract Class	Description
AbsEvent	The abstract class that extends classes that are events.
AbsAssetTaskEvent	The abstract class that extends the `AcknowledgedAssetTaskTranslation` and `RejectedAssetTaskQuote` events.
StartAssetTaskBase	The abstract class that extends the `ApprovedAssetTaskQuote`, `StartAssetTask`, and `StartAssetTaskQuote` events.

9.2.1 AbsEvent Class

Class	AbsEvent
Reference	com.claytablet.model.Event
Implements	IEvent
Direct Known Subclasses	See "AbsEvent Subclasses".
Constructor	AbsEvent ()
Supported Versions	All versions

Important: This is an abstract class, which cannot be instantiated.

This abstract class extends classes that are events.

The AbsEvent class has the following methods:

Return Type	Method and Parameters	Additional Information
AbsEvent	fromXml (System.String eventString)	Deserializes an event object from XML. Returns the deserialized event object. The `event` parameter is the XML string representation of the object.
System.DateTime	getCreateDate ()	Specified by `getCreateDate` in the IEvent interface.
System.String	getEventId ()	Specified by `getEventId` in the IEvent interface.

Return Type	Method and Parameters	Additional Information
System.String	getSourceAccountId ()	Specified by getSourceAccountId in the IEvent interface.
System.String	getTargetAccountId ()	Specified by getTargetAccountId in the IEvent interface.
System.String	removeControlChars (System.String passString)	Replaces ASCII control characters with spaces.
void	setCreateDate (System.DateTime createDate)	Specified by setCreateDate in the IEvent interface.
void	setEventId (System.String eventId)	Specified by setEventId in the IEvent interface.
void	setSourceAccountId (System.String sourceAccountId)	Specified by setSourceAccountId in the IEvent interface.
void	setTargetAccountId (System.String targetAccountId)	Specified by setTargetAccountId in the IEvent interface.
System.String	toXml (AbsEvent eventObject)	Serializes an event object to XML. It returns the XML string representation of the object. The event parameter is the XML string representation of the object.
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by validate in the IEvent interface.

9.2.1.1 AbsEvent Subclasses

The "AbsEvent Class" has the following direct known subclasses:

- AbsAssetTaskEvent
- AcceptAssetTask
- AcceptNeedAssetTaskUpdate

- AcceptNeedTranslationCorrectionAssetTask
- AcceptUpdateTMAsset
- AcknowledgeAssetTaskTranslation
- ApproveAssetTask
- ApproveAssetTaskQuote
- ApprovedAssetTask
- ApprovedTranslationCorrectionAssetTask
- AssetTaskProviderError
- CancelAsset
- CanceledAssetTask
- CanceledSupportAsset
- CancelProject
- CancelSupportAsset
- CompletedProject
- CreateAsset
- CreateSupportAsset
- CreateUpdateTMAsset
- NeedAssetTaskUpdate
- NeedTranslationCorrectionAsset
- ProcessingError
- ReceivedTranslationCorrectionAssetTask
- ReceivedUpdatedAssetTask
- RejectAssetTask
- RejectAssetTaskQuote
- RejectedAssetTask
- ReviewAssetTask
- ReviewTranslationCorrectionAssetTask
- ReviewUpdatedAssetTask
- StartAssetTaskBase
- StartNeedAssetTaskUpdate
- StartNeedTranslationCorrectionAssetTask
- StartSupportAsset
- StartUpdateTMAsset
- SubmitAssetTask
- SubmitProject
- SubmitTranslationCorrectionAssetTask
- SubmitUpdatedAssetTask
- UpdateAssetTaskState
- UpdatedAssetTaskState

9.2.2 AbsAssetTaskEvent Class

Class	AbsAssetTaskEvent
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Direct Known Subclasses	AcknowledgedAssetTaskTranslation, RejectedAssetTaskQuote
Implements	IEvent
Constructor	AbsAssetTaskEvent ()
Supported Versions	All versions

Important: This is an abstract class, which cannot be instantiated.

This abstract class extends the AcknowledgedAssetTaskTranslation and RejectedAssetTaskQuote events.

This class has the following methods:

Return Type	Method and Parameters
System.String	getAssetId ()
System.String	getAssetTaskId ()
System.String	getName ()
System.String	getProjectId ()
System.String	getProjectName ()
System.String	getSourceLanguageCode ()
System.String	getTargetLanguageCode ()
void	setAssetId (System.String assetId)
void	setAssetTaskId (System.String assetTaskId)
void	setName (System.String name)
void	setProjectId (System.String projectId)
void	setProjectName (System.String projectName)

Return Type	Method and Parameters
void	setSourceLanguageCode (System.String sourceLanguageCode)
void	setTargetLanguageCode (System.String targetLanguageCode)

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml
validate

9.2.3 StartAssetTaskBase Class

Class	StartAssetTaskBase
Reference	com.claytablet.model.Event.platform
Extends	com.claytablet.model.Event.AbsEvent
Direct Known Subclasses	StartAssetTask, ApprovedAssetTaskQuote, StartAssetTaskQuote
Implements	IEvent
Constructor	StartAssetTaskBase ()
Supported Versions	All versions

Important: This is an abstract class, which cannot be instantiated.

This abstract class extends the ApprovedAssetTaskQuote, StartAssetTask, and StartAssetTaskQuote events.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
System.String	getAdditionalServicesRequest ()	
System.String	getAssetId ()	
System.String	getAssetTaskId ()	
"ContentType"	getContentType ()	
System.String	getDescription ()	
System.String	getFileExt ()	
"FileType"	getFileType ()	

Return Type	Method and Parameters	Additional Information
IJobMetadata	getJobMetadata ()	
System.String	getName ()	
System.DateTime	getProjectDeadline ()	
System.String	getProjectId ()	
System.String	getProjectName ()	
System.String	getSourceLanguageCode ()	
System.String	getTags ()	
System.String	getTargetLanguageCode ()	
System.boolean	isPendingQuote ()	
void	setAdditionalServicesRequest (System.String additionalServicesRequest)	
void	setAssetId (System.String assetId)	
void	setAssetTaskId (System.String assetTaskId)	
void	setContentType (com.claytablet.model.enm. "ContentType" contentType)	
void	setDescription (System.String description)	
void	setFileExt (System.String fileExt)	
void	setFileType (com.claytablet.model.enm. "FileType" fileType)	
void	setName (System.String name)	
void	setProjectDeadline (System.DateTime projectDeadline)	

Return Type	Method and Parameters	Additional Information
void	setProjectId (System.String projectId)	
void	setProjectName (System.String projectName)	
void	setSourceLanguageCode (System.String sourceLanguageCode)	
void	setTags (System.String tags)	
void	setTargetLanguageCode (System.String targetLanguageCode)	
System.String - If validation passes, this is null. - If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event. Specified by `validate` in the IEvent interface.

This class inherits the following methods from the com.claytablet.model.Event.AbsEvent class:

equals
fromXml
getCreateDate
getEventId
getSourceAccountId
getTargetAccountId
getXStream
hashCode
setCreateDate
setEventId
setSourceAccountId
setTargetAccountId
toString
toXml

9.3 IEvent Interface

Interface	IEvent
Assembly	com.claytablet.model.Event
Known Implementing Classes	See "IEvent Known Implementing Classes" on page 175.
Supported Versions	All versions

This interface defines a collection of method definitions and constant values that are implemented by all classes that are events.

The IEvent interface has the following methods:

Return Type	Method and Parameters	Additional Information
System.DateTime	getCreateDate ()	.
System.String	getEventId ()	
System.String	getSourceAccountId ()	
System.String	getTargetAccountId ()	
void	setCreateDate (System.DateTime createDate)	
void	setEventId (System.String eventId)	
void	setSourceAccountId(System.String sourceAccountId)	
void	setTargetAccountId(System.String targetAccountId)	
System.String ■ If validation passes, this is null. ■ If validation does not pass, the message (String) provides the reason.	validate ()	Validates the fields in this event.

9.3.1 IEvent Known Implementing Classes

The "IEvent Interface" has the following known implementing classes:

- AbsAssetTaskEvent
- AbsEvent
- AcceptAssetTask
- AcceptNeedAssetTaskUpdate
- AcceptNeedTranslationCorrectionAssetTask
- AcceptUpdateTMAsset
- AcknowledgeAssetTaskTranslation
- AcknowledgedAssetTaskTranslation
- ApproveAssetTask
- ApproveAssetTaskQuote
- ApprovedAssetTask
- ApprovedAssetTaskQuote
- ApprovedTranslationCorrectionAssetTask

- AssetTaskProcessingError
- AssetTaskProviderError
- CancelAsset
- CanceledAssetTask
- CanceledSupportAsset
- CancelProject
- CancelSupportAsset
- CompletedProject
- CreateAsset
- CreateSupportAsset
- CreateUpdateTMAsset
- NeedAssetTaskUpdate
- NeedTranslationCorrectionAsset
- ProcessingError
- ReceivedTranslationCorrectionAssetTask
- ReceivedUpdatedAssetTask
- RejectAssetTask
- RejectAssetTaskQuote
- RejectedAssetTask
- RejectedAssetTaskQuote
- ReviewAssetTask
- ReviewTranslationCorrectionAssetTask
- ReviewUpdatedAssetTask
- StartAssetTask
- StartAssetTaskBase
- StartAssetTaskQuote
- StartNeedAssetTaskUpdate
- StartNeedTranslationCorrectionAssetTask
- StartSupportAsset
- StartUpdateTMAsset
- SubmitAssetTask
- SubmitProject
- SubmitTranslationCorrectionAssetTask
- SubmitUpdatedAssetTask
- UpdateAssetTaskState
- UpdatedAssetTaskState

9.4 Language Class

Class	Language
Reference	com.claytablet.model.Language
Constructor	Language ()
Supported Versions	All versions

The `Language` class includes the constants that represent the supported languages and language codes, described in "Language Class Codes". It also contains static methods for working with the language codes, which are described in "Language Class Static Methods" on page 186.

9.4.1 Language Class Codes

The "Language Class" has the following codes:

Type	Language Identifier	Language Code
String	Afrikaans	"af-ZA"
String	Albanian	"sq-AL"
String	Amharic	"am-ET"
String	Arabic_Algeria	"ar-DZ"
String	Arabic_Bahrain	"ar-BH"
String	Arabic_Egypt	"ar-EG"
String	Arabic_Iraq	"ar-IQ"
String	Arabic_Jordan	"ar-JO"
String	Arabic_Kuwait	"ar-KW"
String	Arabic_Lebanon	"ar-LB"
String	Arabic_Libya	"ar-LY"
String	Arabic_MiddleEast	"ar-XR"
String	Arabic_Morocco	"ar-MA"
String	Arabic_Oman	"ar-OM"

Type	Language Identifier	Language Code
String	Arabic_Qatar	"ar-QA"
String	Arabic_Saudi_Arabia	"ar-SA"
String	Arabic_Syria	"ar-SY"
String	Arabic_Tunisia	"ar-TM"
String	Arabic_UAE	"ar-AE"
String	Arabic_Yemen	"ar-YE"
String	Armenian	"hy-AM"
String	Assamese	"as-IN"
String	Basque	"eu-ES"
String	Belarusian	"be-BY"
String	Bengali_Bangladesh	"bn-BD"
String	Bengali_India	"bn-IN"
String	Bosnian_Bosnia_Herzegovina	"bs-BA"
String	Bulgarian	"bg-BG"
String	Burmese	"my-MM"
String	Catalan	"ca-ES"
String	Chinese_Hong_Kong	"zh-HK"
String	Chinese_Macao	"zh-MO"
String	Chinese_PRC	"zh-CN"
String	Chinese_Singapore	"zh-SG"
String	Chinese_Taiwan	"zh-TW"
String	Croatian	"hr-HR"
String	Croatian_Bosnia_Herzegovina	"hr-BA"

Type	Language Identifier	Language Code
String	Czech	"cs-CZ"
String	Danish	"da-DK"
String	Divehi	"dv-MV"
String	Dutch	"nl-NL"
String	Dutch_Belgium	"nl-BE"
String	English_Australia	"en-AU"
String	English_Belize	"en-BZ"
String	English_Canada	"en-CA"
String	English_HongKong	"en-HK"
String	English_India	"en-IN"
String	English_Indonesia	"en-ID"
String	English_Ireland	"en-IE"
String	English_Jamaica	"en-JM"
String	English_Malaysia	"en-MY"
String	English_New_Zealand	"en-NZ"
String	English_Philippines	"en-PH"
String	English_Singapore	"en-SG"
String	English_South_Africa	"en-ZA"
String	English_Trinidad	"en-TT"
String	English_UK	"en-GB"
String	English_US	"en-US"
String	English_Zimbabwe	"en-ZW"
String	Estonian	"et-EE"

Type	Language Identifier	Language Code
String	Faroese	"fo-FO"
String	Farsi	"fa-IR"
String	Filipino	"fil-PH"
String	Finnish	"fi-FI"
String	French	"fr-FR"
String	French_Belgium	"fr-BE"
String	French_Cameroon	"fr-CM"
String	French_Canada	"fr-CA"
String	French_Cote_d_Ivoire	"fr-CI"
String	French_Democratic_Rep_Congo	"fr-CD"
String	French_Haiti	"fr-HT"
String	French_Luxembourg	"fr-LU"
String	French_Mali	"fr-ML"
String	French_Monaco	"fr-MC"
String	French_Morocco	"fr-MA"
String	French_Reunion	"fr-RE"
String	French_Senegal	"fr-SN"
String	French_Switzerland	"fr-CH"
String	Frisian_Netherlands	"fy-NK"
String	Fulfulde_Nigeria	"ff-NG"
String	FYRO_Macedonian	"mk-MK"
String	Gaelic_Ireland	"gd-IE"
String	Gaelic_Scotland	"gd-GB"

Type	Language Identifier	Language Code
String	Gallegan	"gl-ES"
String	Georgian	"ka-GE"
String	German	"de-DE"
String	German_Austria	"de-AT"
String	German_Liechtenstein	"de-LI"
String	German_Luxembourg	"de-LU"
String	German_Switzerland	"de-CH"
String	Greek	"el-GR"
String	Guarani	"gn-PY"
String	Gujarati	"gu-IN"
String	Hausa	"ha-NE"
String	Hawaiian	"haw-US"
String	Hebrew	"he-IL"
String	Hindi	"hi-IN"
String	Hungarian	"hu-HU"
String	Icelandic	"is-IS"
String	Igbo	"ig-NG"
String	Indonesian	"id-ID"
String	Inuktitut	"iu-CA"
String	Italian	"it-IT"
String	Italian_Switzerland	"it-CH"
String	Japanese	"ja-JP"
String	Kannada	"kn-IN"

Type	Language Identifier	Language Code
String	Kanuri	"kr-TD"
String	Kashmiri	"ks-IN"
String	Kazakh	"kk-KZ"
String	Khmer	"km-KH"
String	Konkani	"kok-IN"
String	Korean	"ko-KR"
String	Kyrgyz	"ky-KZ"
String	Lao	"lo-LA"
String	Latin	"la-XL"
String	Latvian	"lv-LV"
String	Lithuanian	"lt-LT"
String	Malay	"ms-MY"
String	Malay_Brunei_Darussalam	"ms-BN"
String	Malayalam	"ml-IN"
String	Maltese	"mt-MT"
String	Maori	"mi-NZ"
String	Marathi	"mr-IN"
String	Mongolian	"mn-MN"
String	Nepali	"ne-NP"
String	Nepali_India	"ne-IN"
String	Norwegian	"nb-NO"
String	Norwegian_Nynorsk	"nn-NO"
String	Oriya	"or-IN"

Type	Language Identifier	Language Code
String	Oromo	"om-ET"
String	Panjabi	"pa-PK"
String	Polish	"pl-PL"
String	Portuguese	"pt-PT"
String	Portuguese_Brazil	"pt-BR"
String	Punjabi_Pakistan	"pa-PK"
String	Pushto	"ps-AF"
String	Quechua_Ecuador	"qu-EC"
String	Quechua_Peru	"qu-PE"
String	Rhaeto_Romance	"rm-IT"
String	Romanian	"ro-RO"
String	Romanian_Moldova	"ro-MD"
String	Russian	"ru-RU"
String	Russian_Moldava	"ru-MD"
String	Sami	"se-NO"
String	Sanskrit	"sa-IN"
String	Serbian_Cyrillic	"sr-RS"
String	Serbian_Latin	"sr-SP"
String	Sindhi_India	"sd-IN"
String	Sindhi_Pakistan	"sd-PK"
String	Sinhala	"si-LK"
String	Slovak	"sk-SK"
String	Slovenian	"sl-SI"

Type	Language Identifier	Language Code
String	Somali	"so-ET"
String	Sorbian	"wen-DE"
String	Spanish	"es-ES"
String	Spanish_Argentina	"es-AR"
String	Spanish_Bolivia	"es-BO"
String	Spanish_Chile	"es-CL"
String	Spanish_Colombia	"es-CO"
String	Spanish_Costa_Rica	"es-CR"
String	Spanish_Dominican_Republic	"es-DO"
String	Spanish_Ecuador	"es-EC"
String	Spanish_El_Salvador	"es-SV"
String	Spanish_Honduras	"es-HN"
String	Spanish_LatinAmerica	"es-XL"
String	Spanish_Mexico	"es-MX"
String	Spanish_Nicaragua	"es-NI"
String	Spanish_Panama	"es-PA"
String	Spanish_Paraguay	"es-PY"
String	Spanish_Peru	"es-PE"
String	Spanish_Puerto_Rico	"es-PR"
String	Spanish_Uruguay	"es-UY"
String	Spanish_US	"es-US"
String	Spanish_Venezuela	"es-VE"
String	Swahili	"sw-TZ"

Type	Language Identifier	Language Code
String	Swedish	"sv-SE"
String	Swedish_Finland	"sv-FI"
String	Syriac	"syr-SY"
String	Tajik	"tg-TJ"
String	Tamil	"ta-IN"
String	Tatar	"tt-RU"
String	Telugu	"te-IN"
String	Thai	"th-TH"
String	Tibetan	"bo-CN"
String	Tigrinya_Eritrea	"ti-ER"
String	Tigrinya_Ethiopia	"ti-ET"
String	Tsonga	"ts-ZA"
String	Tswana	"tn-BW"
String	Turkish	"tr-TR"
String	Turkmen	"tk-TM"
String	Uighur	"ug-CN"
String	Ukrainian	"uk-UA"
String	Urdu	"ur-PK"
String	Urdu_India	"ur-IN"
String	Uzbek	"uz-UZ"
String	Venda	"ve-ZA"
String	Vietnamese	"vi-VN"
String	Welsh	"cy-GB"

Type	Language Identifier	Language Code
String	Xhosa	"xh-ZA"
String	Yi	"ii-CN"
String	Yiddish	"yi-MD"
String	Yoruba	"yo-NG"
String	Zulu	"zu-ZA"

9.4.2 Language Class Static Methods

The "Language Class" has the following static methods:

Return Type	Method and Parameters
List<LanguageCode>	getAddedLanguages ()
System.String[]	getAllLanguages ()
System.String	getFullName (System.String languageName)
int	getLCIDByName (System.String languageName)
System.String	getNameByLCID (int LCID)

9.4.3 LanguageCode Class

Class	LanguageCode
Reference	Event_API.com.claytablet.model.LanguageCode
Constructor	LanguageCode ()
Supported Versions	All versions

The `LanguageCode` class can retrieve and set new language identifiers and codes.

Note: Current language identifiers and codes are constants in the Language class.

The `LanguageCode` class has the following properties:

Return Type	Property
System.String	Code { get; set; }
System.String	Name { get; set; }

9.5 ConnectionContext Class

Class	ConnectionContext
Reference	com.claytablet.model
Implements	StorageDirectoryProvider
Constructors	ConnectionContext () ConnectionContext (bool checkPath)
Supported Versions	All versions

The `ConnectionContext` class holds the connection context fields, such as the local directory paths to use, configuration of the temporary file folders, and connection information between the CMS and TMS.

Recommendation: Use the `app.config` or `web.config` file to configure the connection information, instead of using the ConnectionContext class.

Background: The ConnectionContext class was originally created for the Java SDK. If the ConnectionContext XML file is not in the correct format, serializing and deserializing may throw errors.

The ConnectionContext XML file has the following requirements:

- It must be named called `ConnectionContext.xml`.
- It must be in the folder specified by the `AppSettings["CTT2_ConnectionContext_Folder"]` setting.
- The folder specified by the `AppSettings["CTT2_ConnectionContext_Folder"]` setting should have full permissions for the account your code will run with (especially for the web applications), so that the SDK can save the temporary files.

Note: If you do not create this `ConnectionContext.xml` file, the SDK creates it with the default settings, to prevent some SDK classes throwing errors.

The `ConnectionContext` class has the following methods:

Return Type	Method and Parameters	Additional Information
SortedList<System.String, System.String>	getConnectionParms ()	

Return Type	Method and Parameters	Additional Information
System.String	getSourceDirectory ()	Specified by `getSourceDirectory` in the StorageDirectoryProvider interface.
System.String	getTargetDirectory ()	Specified by `getTargetDirectory` in the StorageDirectoryProvider interface.
System.String	getXmlDataDirectory ()	
void	load ()	Forcibly loads (deserializes) the connection context from an XML file.
void	save ()	Saves (serializes) the connection context to an XML file.
bool	setConnectionContextPath (System.String path)	
void	setConnectionParms (SortedList<System.String, System.String> connectionParms)	

9.6 StorageDirectoryProvider Interface

Interface	StorageDirectoryProvider
Reference	com.claytablet.model
Known Implementing Classes	ConnectionContext
Supported Versions	All versions

The `StorageDirectoryProvider` interface has methods for retrieving the source and target directories.

This interface has the following methods:

Return Type	Method
System.String	getSourceDirectory ()
System.String	getTargetDirectory ()

9.7 Account Class

Class	Account
Reference	com.claytablet.model.Event.Account
Constructor	Account()
Supported Versions	All versions

This class holds account data. The public and private keys are stored as encrypted values. The getters and setters for the two keys automatically handle the encryption and decryption.

This class has the following methods:

Return Type	Method and Parameters	Additional Information
Account	fromXml (System.String xml)	Deserializes an account object from XML.
System.DateTime	getCreateDate ()	
System.String	getEmail ()	
System.String	getId ()	
System.String	getPrivateKey ()	Returns the unencrypted private key.
System.String	getProducerId ()	
System.String	getProviderId ()	
System.String	getPublicKey ()	Returns the unencrypted public key.
System.String	getQueueEndpoint ()	
"QueueType"	getQueueType ()	
System.String	getStorageBucket ()	
"StorageType"	getStorageType ()	
boolean	isActive ()	
void	setActive (boolean active)	

Return Type	Method and Parameters	Additional Information
void	setCreateDate (System.DateTime createDate)	
void	setEmail (System.String email)	
void	setId (System.String id)	
void	setPrivateKey (System.String privateKey)	
void	setProducerId (System.String producerId)	
void	setProviderId (System.String providerId)	
void	setPublicKey (System.String publicKey)	
void	setQueueEndpoint (System.String queueEndpoint)	
void	setQueueType ("QueueType" queueType)	
void	setStorageBucket (System.String storageBucket)	
void	setStorageType ("StorageType" storageType)	
System.String	toXml (Account account)	Serializes an account object to XML.
List<System.String> ■ If validation passes, this is null. ■ If validation does not pass, the list returns messages (String) providing the reasons.	validate ()	Validates the required fields.

9.8 Return Types

Reference	com.claytablet.model.enm

This SDK has the following custom return types:

- "ContentType" on page 191
- "FileType" on page 191
- "QueueType" on page 191
- "StorageType" on page 192

9.8.1 ContentType

The `ContentType` return type describes the type of content that can be submitted for translation. This return type can have the following values:

- Marketing
- Technical
- Legal

9.8.2 FileType

The `FileType` return type describes the type of content file. This return type can have the following values:

- HTML
- Other
- PDF
- Text
- XML

9.8.3 QueueType

The `QueueType` return type describes the implementation of the queue service. This return type can have the following values:

- ActiveMQ
- CT_SQS
- JMS
- Mock
- SQS

9.8.4 StorageType

The `StorageType` return type describes the implementation of the storage device. This return type can have the following values:

- CT_S3
- FS
- FTP
- Mock
- S3

Index