Content Translation Extended

The Content Translation Support Extended (CTSX) module enables you to translate content automatically.
This module comes with several apps which are used to manage the translation tasks.

JIRA	EXCONTRANS
Git	content-translation-ext

Installation

Maven is the easiest way to install the modules. Add the following dependencies to your bundle:

```
<!-- API and base classes that support the CTSX Translator modules. -->
<dependency>
    <groupId>info.magnolia.translation</groupId>
    <artifactId>magnolia-content-translation-support-ext-core</artifactId>
    <version>${CTSExtendedVersion}</version>
</dependency>
```

NOTE: Installation on jdk 11 needs extra dependencies, visit the warnings section of this document.

Translation service

In addition to the core and apps modules, one (or more) of the translator modules should be installed:

- CTSX Google Translator
- CTSX Microsoft Translator
- CTSX Translations.com

Versions

3.1.1	**Magnolia 6.2**
3.0.5	**Magnolia 6.1**
2.9.4	**Magnolia 5.7**
2.9.4	**Magnolia 5.6**

Usage

Content Translation Support Extended (CTSX) enables you to translate content automatically over a translation provider ensuring a more accurate translation. Translation is done from a base language, typically English, to a target language. The base language should exist as content within your website or content app as properties. The target languages should be configured in your site definition. See Enabling multilanguage content for more information.

Configuration

`supportedFieldDefinitions`	**required** Defines the field types that are going to be translated, default is `text` and `richText`, possible values are `composite` and `multiValue`.
`propertiesToTranslateFinder`	**required** The class that finds the properties to be translated, default value `info.magnolia.translation.ext.core.translation.finder.AdaptivePropertiesToTranslateFinder`, finds the properties by processing the fields in the dialogs.
`sourceLanguages`	**optional** • From version 3.0.3 Defines the source languages, the default is the value in the site definition `fallbackLanguage`, you can define as many languages as your 3rd party translator supports, the dialog will display just the ones supported in the current site definition.

config		
supportedFieldDefinitions		
composite	info.magnolia.ui.form.field.definition.CompositeFieldDefinition	String
multivalue	info.magnolia.ui.form.field.definition.MultiValueFieldDefinition	String
richText	info.magnolia.ui.form.field.definition.RichTextFieldDefinition	String
text	info.magnolia.ui.form.field.definition.TextFieldDefinition	String
propertiesToTranslateFinder		
class	info.magnolia.translation.ext.core.translation.finder.AdaptivePropertiesToTranslateFinder	String
sourceLanguages		
de	de	String
en	en	String
es	es	String
pt	pt	String

Translation providers

There are three translation providers to choose from: Google, Microsoft Translator and Translations.com. To make use of the translation functionality, a user account is required for the chosen provider.

A translator module connects Magnolia to translation services. The behavior of the service depends on the translation service selected.

Synchronous	Returns directly the translated result for each translation request. • Google • Microsoft Not support to translate the languages which contain the country information yet. • Translate from English to Germany (Supported) • Translate from English, United Kingdom to Germany, Switzerland (Not supported yet)

Asynchronous	Does not return the translated result for each translation request. A second request is required to get the translated result. • Translations.com Support to translate the languages which contain the country information. • Add the country information for locales in the site definition (e.g: country=GB for English, United Kingdom) • Mix the country information with fallbackLocale (or defaultLocale) property for setting the source language in the site definition (e.g: fallbackLocale=en_GB for English, United Kingdom)

Each translator has a common set of properties.

`autoTranslationAllowed`	**optional** When true the content review process (workflow) is bypassed.
`class`	**required** Definition class for the translator to be used.
`configName`	**required** The display name of the configuration in Magnolia.
`defaultFlag`	**required** Set the translator to be the default.
`enabledFlag`	**required** Enables or disable the configuration with `true` or `false`.
`implementationClass`	**required** Implementation class for the translation service.

Translation Extend app

The module comes with the Translation Extend app which can be found under the tools menu of the appluancher. This app extends the functionality of Content Translation in the sense that translations can now automatized. It would still be possible to export those automatized translations from the Content Translation app after receiving them.

Translation subapps

The app is made up of several subapps for managing translation files and submissions.

- **Batches**: In the Batches subapp users can
  - Add a new or edit an existing batch.
  - Submit a batch for translation.
  - Get the history of the batch.
  - Cancel a batch submission.
  - Reload an exiting batch.
- **Histories**: Allows for viewing and managing the batch histories.
- **Queues**: Allows for managing the translation submission.
- **Comparison**: Allows you to compare between original value and translated value before importing or reject.

Translation status

Once a batch is created it has a status. The status is visible from Batches subapp.

Status	Description
OPEN	Created and not yet submitted
PROCESSING	Processing and waiting to be submitted or imported

SUBMITTED	Submitted to the translation provider
TRANSLATED	Translated by the translation provider and was received
APPROVED	The translation has been approved by the reviewer
IMPORTED	The content was imported
REJECTED	The translation has been rejected by the reviewer
CANCELLED	Cancelled
REOPENED	Reopened after going through review step
EMPTY	The resource has no i18n items

Workflow

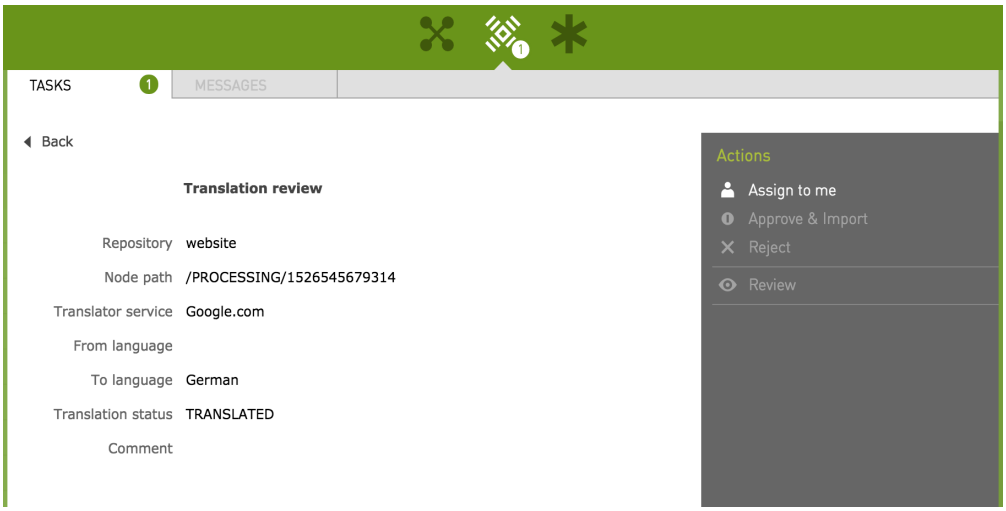
This module provides a workflow for users to review and approve translated content. If the setting `autoTranslationAllowed` is true then the content is imported directly without a review step. Otherwise the content review workflow process is started.

The workflow is identical in nature to the 4-eye content approval workflow used for the Pages app. In this case the editor is the translation provider an the reviewer can be any person in the group. To configure your desired group, change the property here: `/modules/content-translation-support-ext-core/tasks/reviewTranslation/groups@publishers`

Message views

Workflow message views define how messages display in the Pulse during translation workflow. See Message view definition for more.

Message views are configured here: `/modules/content-translation-support-ext-core/messageViews`



Work item handlers

The mapping is based on the work item name which is added to the jBPM process definition. There are two handlers configured for the workflow process.

- **approve**: Handles importing translated content after approval.
- **reject**: Handler for content rejection

Work item handlers are configured here: `/modules/content-translation-support-ext-core/workItemHandlers`

Translation jobs

The core module installs three scheduled jobs to automate the translation process. The translation is submitted to the translation provider. In the case of Translations.com, a second request is required to retrieve the translation results. Finally, the translation is either directly imported or the workflow process is initiated.

The jobs are configured under: `/modules/scheduler/config/jobs`.

Submitting

The `executeSubmissionTranslationCommand` sends translation submission request to provider.

```
# Configuration shown in YAML but must exist as a JCR Node.
executeSubmissionTranslationCommand:
  catalog: workflowTranslation
  command: threadPoolSubmissionCommand
  cron: 1/60 * * * * ? # set to execute every minute at 1 second past the minute
  description: Output, Send, Update translation batch
  enabled: true
  params:
    batchSize: 100 # define how many tuples we retrieve at each execution of the job
    maxAttempt: 10 # define the max retry for each item
    poolSize: 10 # define the size of the thread pool
```

Retrieving

The `executeGetTranslationCommand` is for asynchronous translation. For translating services that require a second request to retrieve the translations you will need to use this job as well.

```
# Configuration shown in YAML but must exist as a JCR Node.
executeGetTranslationCommand:
  catalog: workflowTranslation
  command: threadPoolGetTranslationCommand
  cron: 5/60 * * * * ? # set to execute every minute at 5 seconds past the minute
  description: Get translation from provider
  enabled: true
  params:
    batchSize: 100 # define how many tuples we retrieve at each execution of the job
    maxAttempt: 10 # define the max retry for each item
    poolSize: 10 # define the size of the thread pool
```

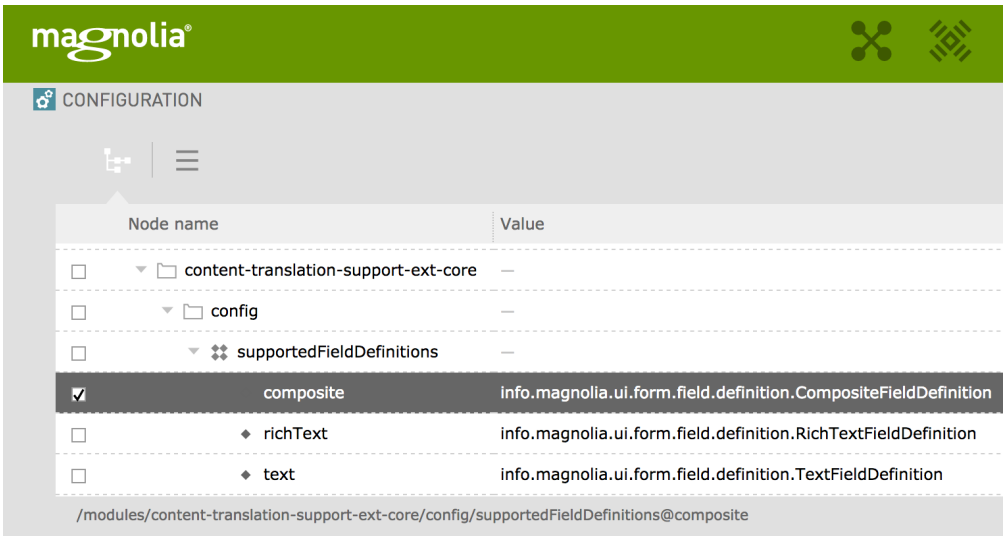
Importing

The `executeImportTranslationCommand` imports the translation immediately or starts the workflow. The behavior is set by `autoTranslationAllo` wed property in the translator configuration.

```
# Configuration shown in YAML but must exist as a JCR Node.
executeImportTranslationCommand:
  catalog: workflowTranslation
  command: threadPoolImportCommand
  cron: 0/30 * * * * ? # set to execute every 30 seconds
  description: Import or create import workflow
  enabled: true
  params:
    batchSize: 100 # define how many tuples we retrieve at each execution of the job
    maxAttempt: 10 # define the max retry for each item
    poolSize: 10 # define the size of the thread pool
```

What is exported and imported

By default, text and rich text fields are included in export files. Additional field types can be registered here: `/modules/content-translation-support-ext-core/config/propertiesToTranslateFinder`. Add another property with the value set to the definition class of the field that should be included.

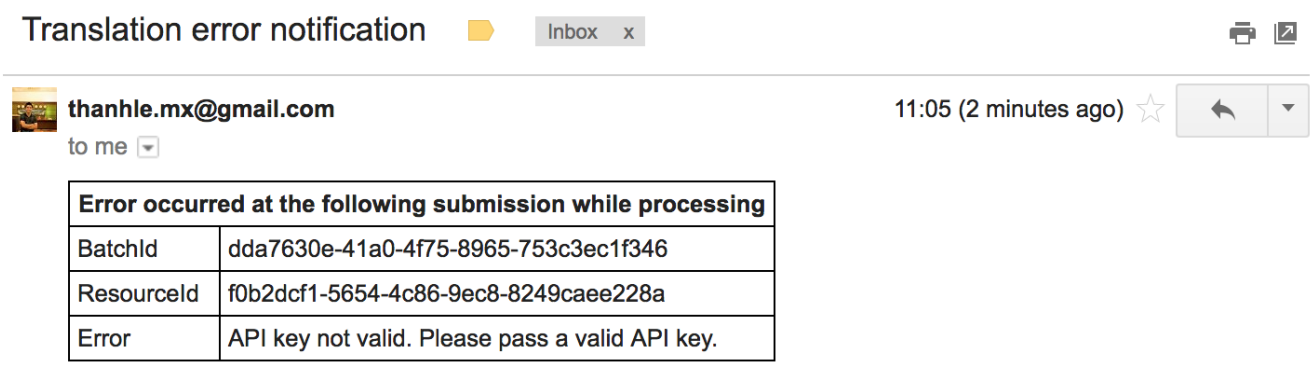


In addition to registration, the child fields of the composite field should have an `i18n` property set to `true`. It is not necessary for the composite field itself to have an `i18n` property.

```
form:
  tabs:
    - name: tabText
      fields:
        - name: composite
          label: Sample Composite
          class: info.magnolia.ui.form.field.definition.CompositeFieldDefinition
          fields:
            - name: title
              i18n: true
              label: Title
              class: info.magnolia.ui.form.field.definition.TextFieldDefinition
            - name: subtitle
              label: Subtitle
              class: info.magnolia.ui.form.field.definition.TextFieldDefinition
```

Notification Email

Magnolia uses Mail module to send a notification email to submitter in case of errors. The template provided is here. The notification email template is configured here: `/modules/mail/config/templatesConfiguration/translationNotificationTemplate`



Warnings

- This module is at INCUBATOR level.
- OpenJDK Java 11 needs additional dependencies.

```
<!-- For OpenJDK 11 CTE 3.0 you will need to have the following dependencies -->
<!-- in your web app lib folder-->

<dependency>
    <groupId>com.sun.xml.bind</groupId>
    <artifactId>jaxb-impl</artifactId>
    <version>2.3.1</version>
</dependency>
<dependency>
    <groupId>javax.xml.ws</groupId>
    <artifactId>jaxws-api</artifactId>
    <version>2.3.1</version>
</dependency>
<dependency>
    <groupId>javax.jws</groupId>
    <artifactId>javax.jws-api</artifactId>
    <version>1.1</version>
</dependency>
<dependency>
    <groupId>com.sun.xml.ws</groupId>
    <artifactId>rt</artifactId>
    <version>2.2.10</version>
</dependency>
<dependency>
    <groupId>com.sun.org.apache.xml.internal</groupId>
    <artifactId>resolver</artifactId>
    <version>20050927</version>
</dependency>
<dependency>
    <groupId>com.sun.xml.stream.buffer</groupId>
    <artifactId>streambuffer</artifactId>
    <version>1.5.6</version>
</dependency>
<dependency>
    <groupId>com.sun.xml.ws</groupId>
    <artifactId>policy</artifactId>
    <version>2.3.1</version>
</dependency>
<dependency>
    <groupId>com.sun.xml.messaging.saaj</groupId>
    <artifactId>saaj-impl</artifactId>
    <version>1.5.0</version>
</dependency>
<dependency>
    <groupId>org.glassfish.gmbal</groupId>
    <artifactId>gmbal-api-only</artifactId>
    <version>3.2.0-b003</version>
</dependency>
```

- The frequency of the scheduled jobs may be high for most. For more information on how to configure the `cron` property see: Scheduler module.
- `DelegatingMultiValueChildNodeWithLocaleTransformer` is the transformer to be used when nesting composite fields in multivalve fields, Magnolia 5.7 and 6.x only.
- There is a `NullPointerException` when translating multilanguage in Magnolia 5.6.6+. Due to changes on 5.6.6, the registered translators were missing after changing language.

  Restarting the server will help to reload the translator. Doing so should help resolve the issue.

- `NoSuchDefinitionException` when clicking on *review* action in translation workflow after upgrading to 2.0.2 There was a missing update of the configuration for `messageViews` node after consolidating translation comparison app with others.

Navigate to: `/modules/content-translation-support-ext-core/messageViews/reviewTranslation/actions/review`

Add properties: `appName=extendContentTranslation` and `subAppName=comparison`

Changelog

- Version 3.1.1
  - Changelog v3.1.1
- Version 3.1.0
  - Changelog v3.1.0
- Version 3.0.5
  - Changelog v3.0.5
- Version 3.0.4
  - Changelog v3.0.4
- Version 3.0.3
  - Changelog v3.0.3
- Version 3.0.2
  - Changelog v3.0.2
- Version 3.0.1
  - Changelog v3.0.1
- Version 3.0
  - Compatible with Magnolia 6.1
  - Changelog v3.0
- Version 2.9.4
  - Compatible with Magnolia 5.6/5.7
  - Changelog v2.9.4
- Version 2.9.3
  - Compatible with Magnolia 5.6/5.7
  - Changelog v2.9.3
- Version 2.9.2
  - Compatible with Magnolia 5.6/5.7
  - Changelog v2.9.2
- Version 2.9.1
  - Compatible with Magnolia 5.6/5.7
  - Changelog v2.9.1
- Version 2.9
  - Compatible with Magnolia 5.6/5.7
  - Changelog v2.9
- Version 2.0.3
  - Changelog v2.0.3
- Version 2.0.2
  - Changelog v2.0.2
- Version 2.0.1
  - Changelog v2.0.1
- Version 2.0 - Initial release of the extensions version of the module.