

# Integrating CMS with XTRF Language Business Platform

## Challenges and Benefits of CMS / TMS Integration

### The challenge

We are ACME Language Services - the translation office - Our customer has a CMS system and frequently orders translations of documents stored in the CMS. CMS operator downloads each document that requires translation and attaches it to an email sent to ACME in which required target languages are specified. A PM at ACME creates a project in the XTRF Home system and begins project processing. ACME applies automated work-flows, CAT integration and automated provider assignment, therefore tedious, manual activities are taken off the shoulders of PMs. When the project is ready an email is automatically sent to the CMS operator informing him/her about the fact and providing file access details. CMS operator downloads ready output files and saves them in the appropriate area within the CMS.

### Current problems

- Most of the time files and project details are sent by e-mail. Manual activity both on the customer and ACME side is required - a great amount of CMS operator and PM time gets wasted.
- CMS does not store information whether a translation has already been ordered for a given document therefore the CMS operator sometimes requests projects for the same document twice.
- There usually are dozens of translation projects opened at a given time - our CMS operator spends a few minutes on a manual upload of each translated document as it is hard to find the right area for a document.

### Opportunity

An IT geek familiar with the CMS engine is eager to date the CMS operator and is willing to devote a few evenings on implementation of a solution that would radically decrease her workload....

### Expected benefits

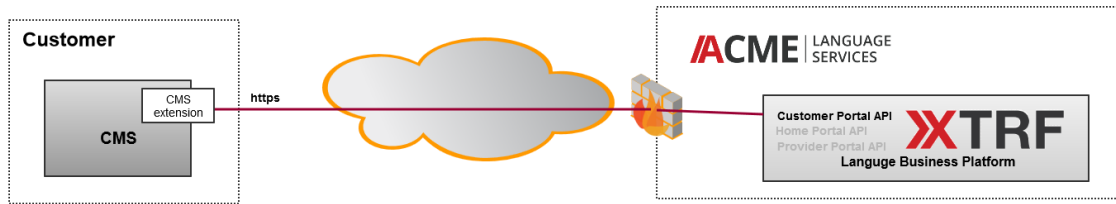
#### Customer benefits

1. Client working in his preferred CMS through integration may easily trigger projects in XTRF. In this respect CMS acts as the XTRF Client Portal.
2. No need to learn how to use yet another external system
3. No need to memorize another set of login credentials
4. Client continues working in a very well familiar environment
5. Client just marks text area that must be translated, presses button and instantly the request is transferred to XTRF, project is created and a PM is notified
6. Full project automation possible
7. No need to manually create request and send it to the translation company

#### ACME benefits

1. Continuous flow of incoming projects
2. Streamlined communication with end client
3. PMs do not have to create projects manually
4. PM efficiency increased through reduction of manual activities
5. Ability to relocate PM time to more challenging tasks
6. An obvious benefit of the IT Geek: the date...

## Solution Architecture



The CMS of the end client will be extended with a dedicated module capable of communicating with the XTRF Language Business Platform used by ACME. For the sake of integration REST based XTRF Customer Portal API and secure communication channel over https protocol will be used.

## Use cases

The CMS operator expects a few additional activity types available directly in the CMS system.

### Creation of a translation project

A basic action provided by the CMS extension module allows to 'order translation'. This action should be available for the CMS operator in the context of a CMS document (e.g. as a menu or context menu extension).

When the CMS operator decides to order a translation of a given document through CMS, she clicks the button and confirmation window appears where she specifies source and target languages as well as other optional project properties (e.g. deadline, specialization, workflow). By pressing the 'confirm' button the operator triggers the following action: CMS extension calls ACME customer's portal API, and passes the document file together with project properties. A project is created in ACME's XTRF installation and is immediately visible at the ACME's Project Manager dashboard.

### Checking translation project status

The CMS engine periodically monitors project status and checks if it is ready. The IT Geek uses a unique customer project id that he has assigned to the project upon creation in order to locate it and verify its current status in ACME.

### Fetching translated files

Project status verification activity reported that the project is ready. The IT Geek downloads all the result files from ACME as zip. He places files in appropriate area of the CMS structure.

## Customer Portal API

To support implementation of the actions described above the IT Geek will use XTRF Customer Portal API. He has found the API documentation for the XTRF Language Business Platform at [Welcome to XTRF API Knowledge Base without any problem.](#)  
 //the REST API is not public yet so please use the pdf version/

### Creation of a translation project

The IT Geek implemented the entire code required by the CMS system to extend its functionality with the 'send for translation' action and finally arrived at the moment when the use of XTRF Customer Portal API is necessary. He followed the action sequence below:

1. Obtain sessionId - by using [POST] /system/login - pass username and password used by the CMS operator to login to Customer Portal. Please not that main customer profile credentials may be used as far as any contact person credentials of the customer. IT Geek created a dedicated, artificial contact person account to be used for the integration purposes and check if he could access the portal using the browser (using <http://xtrf.instance.url/customers/>) to validate if user name and password provided by the CMS operator are valid)
2. Set http header property Cookie: JSESSIONID=<session\_id> for all further requests
3. Prepare project creation request
  - a. [POST] /system/session/files - is used to send files, it returns an id of the uploaded file - the IT Geek used this

- method twice to send the document to be translated and a reference file as well,
- b. [POST] /quotes - is used for both project and quote creation -
    - i. as the IT Geek wants to create a project rather than a quote he has set JSON property "autoAccept" as correct.
    - ii. in order to be able to easily reference the project, the IT Geek has generated a uniqueProjectNumber and set it as a value of the "customerProjectNumber" property.
    - iii. the final JSON request includes more properties but having read the documentation and following a discussion with the CMS operator the IT Geek had no more doubts about them.

```
{
  "name" : "Google Gloves",
  "customerProjectNumber" : uniqueProjectNumber,
  "workflow" : { "name" : "TP" },
  "specialization" : { "name" : "Economy" },
  "sourceLanguage" : { "name" : "English" },
  "targetLanguages" : [ { "name" : "German" } ],
  "deliveryDate" : { "formatted" : "2014-09-15 11:30:00" },
  "notes" : "Sample notes",
  "autoAccept" : true,
  "person" : { "id" : 402 },
  "files" : [ { "id" : 1415596305 }, { "id" : 2005194325 } ],
  "referenceFiles" : [ { "id" : 4129771301 } ],
}
```

- /system/logout used to logout. - the IT Geek is very well aware of security issues therefore he always remembers to close the http session and to invalidate the cookie

## Checking translation project status

The CMS engine checks periodically whether the project is ready, using the following methods:

1. login and session Id as in the previous example
2. [GET] /projects?customerProjectNumber=uniqueProjectNumber
  - a. method call returns a list of projects matching the query criteria
    - i. the list will contain each project only provided that the corresponding quote (created in the previous step) has already been accepted (converted to a project) by an ACME Project Manager.
    - ii. ACME may decide not to validate the customer quote manually but rather to enable automated quote-to-project conversion in the customer profile in ACME's XTRF - see details at XTRF Help [Advanced - Add / Edit Price Profile - Customer](#).
  - b. Project status as well as some other project details are provided.

```
[ {
  "id" : 576,
  "idNumber" : "2012/57",
  "refNumber" : "P1712",
  "status" : "OPENED",
  "projectManager" : {
    "id" : 1,
    "type" : "User",
    "name" : "Tomasz Mróz",
    "position" : "Project Manager"
  },
  "startDate" : "2012-08-07 09:00:00",
  "deadline" : "2012-08-28 17:00:00",
  "awaitingCustomerReview" : true,
  "totalAgreed" : {
    "amount" : 800.00,
    "currency" : "EUR"
  }
}
```

3. logout

## Fetching translated files

When the project is moved to the "Closed" status, the translated files are ready to be downloaded.

1. login and session Id as in the previous example.
2. [GET] /projects?customerProjectNumber=[uniqueProjectNumber]
  - a. extract project 'id' property from the JSON structure.
  - b. use [GET] /projects/{id}/files/outputFilesAsZip to open a stream to zip file containing output files from a given project.
  - c. store the translated file in the appropriate area of the CMS.
3. logout.

## Code Sample

Live demo site: <http://xtrf.github.io/cp-api/>

Code sample: <https://github.com/xtrf/cp-api>

## Preparing test environment

It Geek does not like to loose his valuable time so he spent few minutes on preparing test environment before executing the integration tests. He went through the following points:

- Create an artificial customer named 'test' with Geek's email address and random password
- Prepare a 'returnInputAsOutput' workflow. To set up the automatic action used by the workflow you will need administrator / root access to the server where XTRF is installed.
  1. Go to System > Configuration > Workflows > Automatic Actions.
  2. Click Add.
  3. Select External Process as the Type and enter Do Nothing into the Name field. Save your changes. The automatic action is created.
  4. Edit the action you have created. A warning message is displayed: External processes editing is not permitted. If you are aware of the security risk and want to enable external processes please recreate a file <file\_path> on the XTRF server and refresh this page.

5. Log in to the the XTRF server as administrator / root and recreate the file indicated in the warning message. You are now temporarily allowed to set up external processes in XTRF.
6. Edit the automated action you have created. In the section Standard Output File enter ls.out.txt. In the field Exit Code Actions set the following values:

	Exit Code	Comparison	Action
Exit Code Actions	<input type="text" value="0"/>	Unspecified ▼	Synchronise Output Files
	<input type="text" value="0"/>	Unspecified ▼	Finish Workflow Step

7. Click Save&Exit. Go to Workflow Definitions and click Add.
8. In the Workflow Graph tab of the workflow definition, add a new job in the Project jobs section. A new window pops up.
9. In the Main Settings tab of the Workflow Job window, select Default Provider > Automatic Action > Do Nothing. Leave the other fields as default.
10. In the Files tab of the Workflow Job window, select Copy Missing Output Files from Input on Finish. Save your changes. The pop-up window closes.
11. In the Quote tab of the workflow definition, select Automatically Convert Quote Accepted by Customer and Start Workflow.
12. In the Receivables tab of the workflow definition, click Add Receivable. A new window pops up. See below for an example of the settings:

**Task Receivable** ? 🗑️ ✕

Job Type	<input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="translation"/>
Calculation Unit	<input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="source word"/>
Quantity	<input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="10,000"/>
Discount / Surcharge	<input type="button" value="Edit"/>
Description	<input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="aPI call"/>

- 13.
14. Save your changes and exit the workflow definition.
  - set it as the only one used by customer test (in customer profile 'Portal Settings' tab in XTRF Home Portal)
  - use XTRF Customer Portal to login as 'test' customer 'place an order' and make sure the order finalize automatically delivering input files as the result (an email with the link to the file sis expected).

## Future extension and modifications

The entire functional range described above forms a relatively simple example. In a real-life scenario one should consider the at least the following aspects:

- structure and capabilities of CMS extension - eg. storing selected configuration elements such as login/password to XTRF Customer Portal API. Thus the CMS operator does not have to provide them each time - if there are multiple CMS operators, should they use the same, common credentials for project creation in ACME?,
- document versioning on the CMS side enables customer to store information which version of a document has

actually been translated - if document flow requires such support a few hours with the CMS operator need to be spent in order to validate the required flow that should be developed.

- where translated files need to be stored - it is essential to point to the correct location - this aspect is a challenging one as in the majority of cases input and output locations are different (patterns of paths for the storage of translated documents).
- multiple target languages - if a document is expected to be translated into multiple languages it will be necessary to enter more information into project structure with the GET /projects/{projectId} method. It will then be possible to trace the status of each target language independently and to finally fetch the output files as individual zips for each language combination using GET /projects/tasks/{taskId}/files/outputFilesAsZip