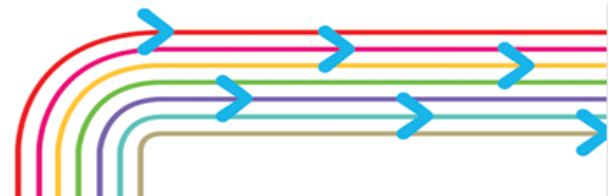




Integration Web Services v2



Documentation for XTM Integration Web Services version 2

Published by XTM International Ltd.

© Copyright XTM International Ltd. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, including photocopying, without the written permission of XTM International Ltd.

Updated April 2012



XTM-International Ltd, PO Box 2167, Gerrards Cross, SL9 8XF, UK
Tel.: +44 (0)1753 480479 email: sales@xtm-intl.com <http://www.xtm-intl.com>

1. Table of Contents

1.	TABLE OF CONTENTS	3
2.	REVISION HISTORY	4
3.	THE BENEFITS OF INTEGRATING XTM USING XTM WEB SERVICES	5
4.	OVERVIEW OF XTM WEB SERVICES	6
5.	DEFINITIONS	8
6.	ROLES AND WEB SERVICE ACCESS	10
7.	CREATE USERS	11
8.	CREATE CUSTOMERS.....	12
9.	CREATE CUSTOMER PROJECT MANAGERS	13
10.	CREATE A PROJECT 1	14
10.1.	SPECIAL CHARACTERS IN THE FILE URL	17
11.	CREATE A PROJECT 2	18
12.	MANAGE THE PROJECT WORKFLOW.....	19
13.	CHECK THE STATUS OF PROJECTS AND JOBS	20
14.	OBTAIN PROJECT METRICS AND STATISTICS	21
15.	GENERATE, CHECK THE STATUS AND DOWNLOAD FILES.....	22
16.	DOWNLOAD FILES.....	23
17.	OBTAIN ADDITIONAL DATA	24
18.	UPLOAD AN XLIFF FILE	25
19.	OPEN XTM MODULES	26
20.	PHP EXAMPLES.....	28
20.1.	WEB SERVICE FRAMEWORK FOR PHP INSTALLATION GUIDE	28
20.2.	CREATEPROJECTURL.....	29
20.3.	CHECKPROJECTANALYSISCOMPLETION.....	30
21.	JAVA EXAMPLES.....	31

2. Revision history

Date	Changes
12/04/2012	• Document created.

3. The Benefits of Integrating XTM Using XTM Web Services

XTM Web services have been designed to allow users to easily integrate XTM with existing CMS or project management systems.

- XTM Web services offer complete flexibility to suit the way you wish to work. For example:
 - XTM can be treated as a translation “black box” that handles all the issues related to translation such as resource allocation and workflow.
 - The workflow, translation resources and other parameters can be defined in the third party system and then sent to XTM with the documents for translation. Status updates and reports including project metrics can then be obtained from XTM.
- XTM Web services reduce the time and effort required for the integration to a minimum.
- XTM can be integrated with any system as long as it supports Web services. It is not important what programming language has been used to write the third party system.
- Updates to XTM will be compatible with XTM Web services. This means you will have instant access to the latest features in XTM as soon as they are developed by XTM International.
- The integration of XTM into your project or translation workflow streamlines processes, speeds throughput times and assists project management.
- XTM Web services are available for XTM Suite and XTM Cloud.

4. Overview of XTM Web Services

If you wish to test to use the XTM Web Services please contact sales@xtm-intl.com who can provide you with the requisite URLs.

The XTM Web service is designed to be completely passive. It does not know anything about the Customer service and does not call the Customer service at all. It only responds to calls from the customer.

For file transfers, the XTM web service can accept a URL or an MTOM file attachment.

There are two key concepts for the XTM web service:

1. Project - this is the overall project that is created.
2. Job - a Project may comprise one or more language pairs (source language and target language). Each language pair is called a 'job' within a project.

The following steps describe how to use the XTM web service:

1. Create a project. Certain data is mandatory to create a project and other data is optional.

Mandatory data

- Source files
- Source language
- Target languages
- Customer

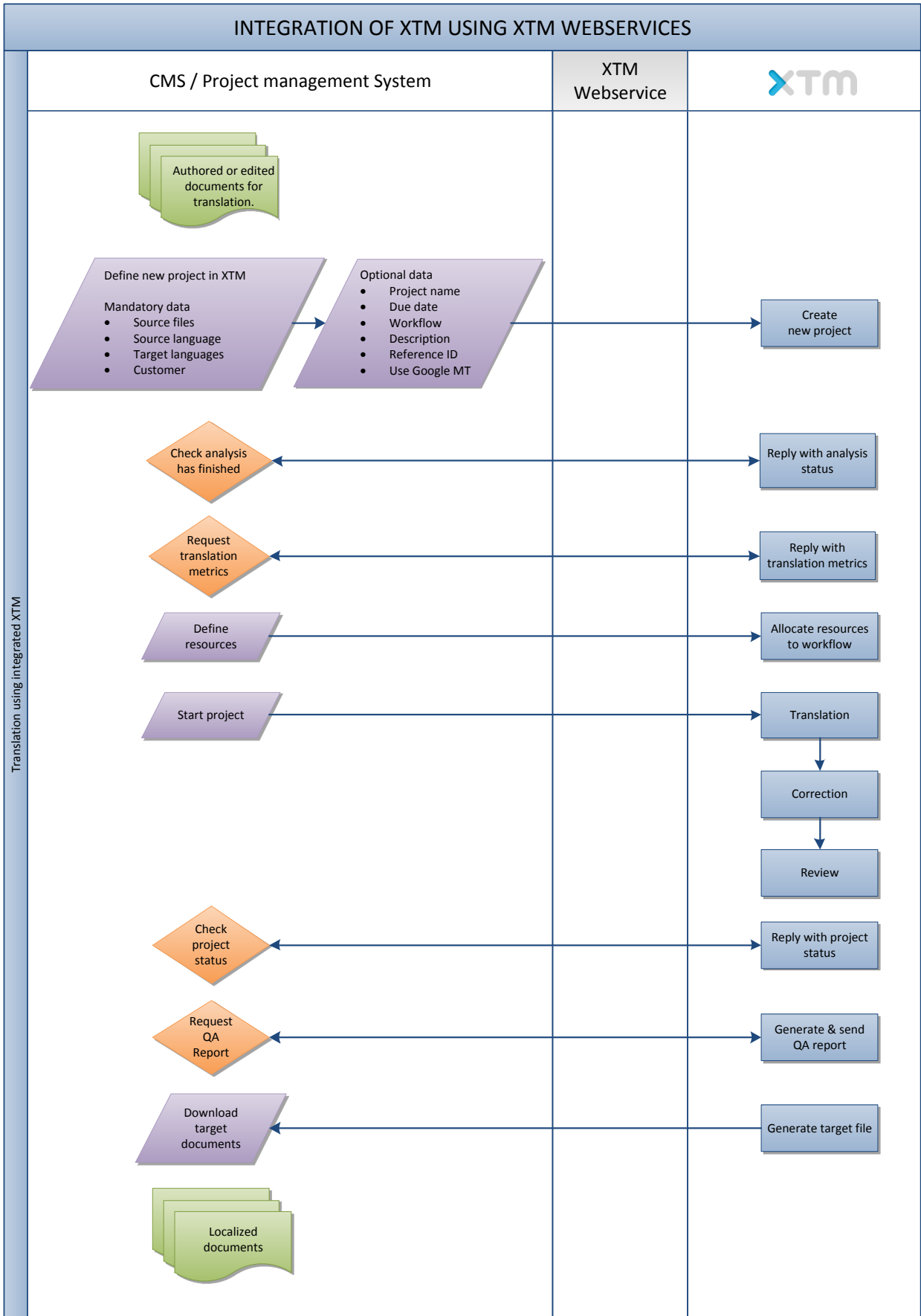
Optional data

- Project name
- Due date
- Workflow
- Description
- Reference ID
- Use Google MT

2. Check if the project/or particular job has finished its analysis stage
3. Get the job/project metrics if required. Depending on the metrics the Customer may decide not to proceed with the project.
4. Assign linguists.
5. Start or delete the Project.
6. Check the status of the Project.
7. Generate the target file, XLIFF or QA report and download it (you can either request URL or MTOM download).

Steps 2. to 5. are optional since the project manager can perform these tasks from within XTM.

The diagram below shows a typical integration.



Translation using integrated XTM

Integration of XTM using XTM Web services

5. Definitions

1. Definitions of common objects

- **LoginAPI:** userId, password, client
Required for all web service calls.
- **XTMCustomerDescriptorAPI:** id, name, externalId
The Customer descriptor is used to specify a customer. Firstly the customer ID is checked, then the customer name, finally the customer external ID.
- **XTMDomainDescriptorAPI:** domain, domainName
The Domain descriptor is used to specify a domain. Firstly domain definition from XTM_DOMAINS enumeration type is checked, then the custom domain name.
- **XTMFileDescriptorAPI:** id
The File descriptor is used to specify a file.
- **XTMJobDescriptorAPI:** id, externalId
The Job descriptor is used to specify a job. Firstly the job ID is checked, then the job external ID.
- **XTMProjectDescriptorAPI:** id, externalId
The Project descriptor is used to specify a project. Firstly the project ID is checked, then the project external ID.
- **XTMUserBaseDescriptorAPI:** id, name
The Base user descriptor is used to specify a user. Firstly the user ID is checked, then the user name.
- **XTMUserDescriptorAPI:** id, name, actorType
User descriptor is used to specify a linguist or LSP. Firstly the user ID is checked, then the user name. Actor type allows to specify linguist or LSP.
- **XTMWorkflowDescriptorAPI:** workflow, workflowName
Workflow descriptor is used to specify a workflow. Firstly the workflow definition from XTM_WORKFLOWS enumeration type is checked, then the custom workflow name.
- **XTMWorkflowStepDescriptorAPI:** workflowStep, workflowStepName
Workflow step descriptor is used to specify a workflow step. Firstly workflow step definition from XTM_WORKFLOW_STEP enumeration type is checked, then the custom workflow step name.

2. Enumeration types

- XTM_WORKFLOW_STEP - list of possible workflow steps

```
/** define for 'translate 1' step. */  
TRANSLATE1,
```

```
/** define for 'review 1' step. */  
REVIEW1,
```

```
/** define for 'review 2' step. */  
REVIEW2,
```

```
/** define for 'correct 1' step. */  
CORRECT1,
```

```
/** define for 'correct 2' step.*/  
CORRECT2;
```

- ActorType - list of possible linguist types

```
/** define for linguist type. Default value if not set. */  
INTERNALLINGUIST,
```

```
/** define for LSP type. */  
LSP
```

- LANGUAGE_CODE

Contains languages in the IANA format where a single language subtag composed of two letters is followed by a region subtag composed of two letters, for example: en_GB, pl_PL, ru_RU.

6. Roles and Web Service access

The LoginAPI object is required for all calls.

Access to the various methods in the XTM Web Services is controlled by the logon credentials. There are three roles that reflect the functionality in the GUI of the application

ROLE	AVAILABLE WEB SERVICE FUNCTIONALITY	WEBSERVICE
Administrator	Create users Obtain the URL to translation properties of system and customer	Project Manager web service
Internal project manager	Create users, customers and projects Create customer project managers Select which TM to use while creating project Obtain project metrics Manage the workflow Check the status of the project Obtain additional information about the project Download files Upload an XLIFF file Obtain the URL of XTM Editor for a specific XLIFF file Obtain translator statistics Obtain the URL to Project Editor Obtain base information about XTM Obtain list of supported file types Set project activity Obtain the URL to translation properties of customer	Project Manager web service
Customer project manager	Create projects Obtain project metrics Manage the workflow Check the status of the project Obtain additional information about the project Download files Obtain translator statistics Obtain the URL to Project Editor Obtain base information about XTM Obtain list of supported file types Set project activity	Customer web service

7. Create Users

1. createUser()

A user can be created by specifying the XTMUserAdditionAPI object. The information is divided into the following groups:

- user details (contains also: address details and instant messaging identifiers)
- roles
- language combinations
- qualifications
- domains
- terminology rights

Most of the fields are not mandatory however the following must be specified:

- user details
 - username
 - password
 - first name
 - last name
 - e-mail address
- roles

Language Combinations are specified by specifying the Source language and Target language using language codes.

USER_ADDITION_ROLE_API – list of possible roles

- TRANSLATOR
Additionally Language Combinations group is mandatory.
- REVIEWER
Additionally Language Combinations group is mandatory.
- CORRECTOR
Additionally Language Combinations group is mandatory.
- TERMINOLOGY_EXPERT
Additionally Terminology rights group is mandatory. If terminology expert is not a global expert then term customer descriptor is also required.
- PROJECT_MANAGER
- TM_EXPERT

8. Create Customers

1. createCustomer()

A customer can be created by specifying the XTMCustomerAPI object. The information is divided into the following groups:

- customer base
- address details
- instant messaging identifiers
- additional settings

Most of the fields are not mandatory however the following must be specified:

- customer base
 - name
 - external customer descriptor – optional

XTMExternalCustomerDescriptorAPI: externalId

External customer descriptor needs to be specified to use customer external ID in other methods instead of customer ID from XTM.

9. Create Customer Project Managers

1. createCustomerPM()

A Customer Project Manager can be created by specifying the XTMCustomerPMAPI object. The information is divided into the following groups:

- user details (contains also: address details and instant messaging identifiers)
- customer descriptor
- terminology rights

Most of the fields are not mandatory however the following must be specified:

- user details
 - username
 - password
 - first name
 - last name
 - e-mail address
- customer descriptor

CustomerAdminRoleAPI - possible customer project manager roles

```
/** Manager role. */
MANAGER,
```

```
/** Viewer role. */
VIEWER;
```

WorkflowAccessLevelAPI - possible workflow access levels

```
/** Trusted level. */
TRUSTED,
```

```
/** Standard level. */
STANDARD,
```

```
/** Minimal level. */
MINIMAL;
```

TermRightAPI - possible terminology rights

```
/** Export right. */
EXPORT,
```

```
/** Import right. */
IMPORT,
```

```
/** Add right. */
ADD,
```

```
/** View right. */
VIEW,
```

```
/** Modify right. */
MODIFY;
```

10. Create a project 1

There are two ways of creating a project:

1. createProjectURL() - based on file URL, requires XTMPProjectURLAPI object
2. createProjectMTOM() - based on MTOM objects, requires XTMPProjectMTOMAPI object

XTMPProjectBaseAPI - the base project information:

```

/** Name of the project. (Optional - if not present will be auto generated) */
private String name;

/** Source language. (Mandatory.) */
private LANGUAGE_CODE sourceLanguage;

/** Target languages. (Mandatory, at least one must be present) */
private List<LANGUAGE_CODE> targetLanguages;

/** Customer descriptor. */
private XTMCustomerDescriptorAPI customer;

/** Due date. (Optional, if missing the current date is assumed) */
private Date dueDate;

/** Domain descriptor. */
private XTMDomainDescriptorAPI domain;

/** Workflow descriptor. (Optional, if missing 'Translate' is assumed) */
private XTMWorkflowDescriptorAPI workflow;

/** External project descriptor. (Optional) */
private XTMEExternalProjectDescriptorAPI externalDescriptor;

```

XTMPProjectAPI extends XTMPProjectBaseAPI – the extended project information:

```

/** Additional description. (Optional) */
private String description;

/** The reference Id. (Optional) */
private String referenceId;

/** Info if the linguists should have possibility to use google machine translation help. (Optional)
 */
private Boolean useGoogleMachineTranslation;

/** Info if not approved translation memory should be used. (Optional) */
private Boolean usesNotApprovedTranslationMemory;

/** Allows editing ICE matches. (Optional) */
private Boolean allowEditingOfICEMatches;

/** Specifies in which steps non translatables should be marked as done. (Optional) */

```

```
private WorkflowStepUsageEnum nonTransAsDONE;
```

```
/** Specifies in which steps leverages should be marked as done. (Optional) */
private WorkflowStepUsageEnum leveragesAsDONE;
```

XTMProjectURLAPI extends XTMProjectAPI:

```
/** Translation files. (files requiring translation. Mandatory, at least one must be present) */
private List<XTMFileURLAPI> translationFiles;
```

```
/** Reference Material files. (Optional) */
private List<XTMFileURLAPI> materialFiles;
```

XTMProjectMTOMAPI extends XTMProjectAPI:

```
/** Translation files. (files requiring translation. Mandatory, at least one must be present) */
private List<XTMFileMTOMAPI> translationFiles;
```

```
/** Reference Material files. (Optional)*/
private List<XTMFileMTOMAPI> materialFiles;
```

(<https://cwiki.apache.org/CXF20DOC/mtom-attachments-with-jaxb.html>)

XTMFileURLAPI and XTMFileMTOMAPI extend XTMFileAPI which contains:

```
/** File name. */
private String filename;
```

```
/** External job descriptors. It maps external ID to the combination of the source file and
language code. */
```

```
private Map<LANGUAGE_CODE, XTMEExternalJobDescriptorAPI> externalDescriptors;
```

Example:

File for translation: simple.txt

Target languages: pl_PL, ru_RU

In this case 2 jobs will be created in XTM. External descriptors create the possibility to assign an external ID for each job (file with language pair).

XTMEExternalProjectDescriptorAPI: externalId

External project descriptor needs to be specified to use project external ID in other methods instead of project ID from XTM.

XTMEExternalJobDescriptorAPI: externalId

External job descriptor needs to be specified to use job external ID in other methods instead of job ID from XTM.

Each creation method returns a XTMProjectResponseAPI object which includes:

- project descriptor
- name
- jobs – list of XTMJobResponseAPI objects which contain:
 - job descriptor
 - file name
 - source language
 - target language

XTM_WORKFLOWS - possible XTM general Workflows

```

/** Define for 'translate'. */
TRANSLATE,

/** Define for 'translate # review'. */
TRANSLATE_F_REVIEW,

/** Define for 'translate # review # review'. */
TRANSLATE_F_REVIEW_F_REVIEW,

/** Define for 'translate -> correct'. */
TRANSLATE_P_CORRECT,

/** Define for 'translate -> correct -> correct'. */
TRANSLATE_P_CORRECT_P_CORRECT,

/** Define for 'translate , correct'. */
TRANSLATE_CORRECT,

/** Define for 'translate , review'. */
TRANSLATE_REVIEW,

/** Define for 'translate , review -> review'. */
TRANSLATE_REVIEW_P_REVIEW,

/** Define for 'translate , correct -> review'. */
TRANSLATE_CORRECT_P_REVIEW,

/** Define for 'translate # correct'. */
TRANSLATE_F_CORRECT,

/** Define for 'translate -> correct # review'. */
TRANSLATE_P_CORRECT_F_REVIEW;

/** Define for 'translate # correct # review'. */
TRANSLATE_F_CORRECT_F_REVIEW;

```

WorkflowStepUsageEnum - possible steps where leverages and non translatables can be marked as done

```

/** None. */
NONE,

/** In all steps. */
ALL_STEPS,

/** Only in the first step. */
FIRST_STEP

/** All steps except the last. */
NOT_LAST_STEP;

```

10.1. Special characters in the file URL

If a translation file URL contains one of the specific character below, then an additional URL transformation is required to make the file available for the Web Service. The table below shows how specific characters should be processed.

Character	Encoded character	URL example
	%20	http://test%20.xml
`	%60	http://test%60.xml
~	~	http://test~.xml
!	!	http://test!.xml
@	@	http://test@.xml
#	%23	http://test%23.xml
\$	\$	http://test\$.xml
%	%25	http://test%25.xml
^	%5E	http://test%5E.xml
&	%26	http://test%26.xml
((http://test(.xml
))	http://test).xml
-	-	http://test-.xml
_	_	http://test_.xml
=	=	http://test=.xml
+	+	http://test+.xml
{	%7B	http://test%7B.xml
}	%7D	http://test%7D.xml
[%5B	http://test%5B.xml
]	%5D	http://test%5D.xml
;	;	http://test;.xml
'	'	http://test'.xml
,	,	http://test,.xml
.	.	http://test..xml
*	Forbidden	
:	Forbidden	
“	Forbidden	
<	Forbidden	
>	Forbidden	
?	Forbidden	
/	Forbidden	
	Forbidden	
\	Forbidden	

11. Create a Project 2

This method is only available to in-house project managers. It includes all the settings available in Create a project 1 but also permits additional settings.

1. `createProjectForPMURL()`
Creates a project with additional settings only available to in-house Project Managers. Based on file URL, requires `XTMProjectForPMURLAPI`.
2. `createProjectForPMMTOM()`
Creates a project with additional settings only available to in-house Project Managers. Based on MTOM objects, requires `XTMProjectForPMMTOMAPI`.

Additional settings:

- list of TM Customers (`XTMCustomerDescriptorAPI` objects)
This provides the possibility to specify that the project should use the TM of multiple customers. This option requires customer descriptors.

12. Manage the Project Workflow

1. `assignLinguistToProject()`
Assigns linguists or LSPs to all jobs (all bundles, all files and all languages) for a specific workflow step in the project. It requires project descriptor and a list of `XTMStepLinguistAssignmentAPI` objects.

The `XTMStepLinguistAssignmentAPI` object includes workflow step descriptor and user descriptor.
2. `assignLinguistToJob()`
Assigns linguists or LSPs to the specified jobs. It requires a list of `XTMJobLinguistAssignmentAPI` objects.

The `XTMJobLinguistAssignmentAPI` object includes job descriptor and a list of `XTMStepLinguistAssignmentAPI` objects which are described in the point 1.
3. `startProject()`
Starts the given projects. It requires a list of projects descriptors.
All jobs in the projects are started and linguists receive emails about their tasks.
4. `updateProjectActivity()`
It allows to delete, activate or archive the specified projects. It requires a list of projects descriptors and activity.

When assigning linguists or starting a project on a LSP subcontractor's system, it is necessary to obtain the original project ID from the main XTM instance and then use the subcontractor's web service with the original project ID as the external project ID in this method.

13. Check the Status of projects and jobs

- 1) `checkProjectAnalysisCompletion()`
Checks if analysis of the project and its jobs is completed. This method requires project descriptor and returns the general project status (`XTM_PROJECT_COMPLETION_STATUS`) and status for each job (`XTM_JOB_COMPLETION_STATUS`).
- 2) `checkJobAnalysisCompletion()`
Checks if analysis of the job is completed. This method requires a list of jobs descriptors and returns the status for each job (`XTM_JOB_COMPLETION_STATUS`).
- 3) `checkProjectCompletion()`
Checks if project is completed. This method requires project descriptor and returns the general project status (`XTM_PROJECT_COMPLETION_STATUS`) and status for each job (`XTM_JOB_COMPLETION_STATUS`).
- 4) `checkJobCompletion()`
Checks if job is completed. This method requires a list of job descriptors and returns the status for each job (`XTM_JOB_COMPLETION_STATUS`).

`XTM_PROJECT_COMPLETION_STATUS` - possible project completion statuses

```
/** All jobs in the given project are still processing. */
IN_PROGRESS,
```

```
/** At least one job in the given project is completed. */
PARTIALLY_FINISHED,
```

```
/** All jobs in the given project are completed. */
FINISHED;
```

`XTM_JOB_COMPLETION_STATUS` - possible job completion statuses

```
/** A job is still processing. */
IN_PROGRESS,
```

```
/** A job is completed successfully. */
FINISHED,
```

```
/** A job could not be completed. */
ERROR;
```

14. Obtain Project Metrics and Statistics

- 1) `obtainProjectMetrics()`
Obtains original and up to date metrics for all jobs and general metrics for the whole project for each target language or one specified target language. This method requires project descriptor and optional target language in the additional method options.
- 2) `obtainJobMetrics()`
Obtains original and up to date metrics for the specified jobs. This method requires a list of jobs descriptors.
- 3) `downloadProjectMetricsURL()`
Downloads the XLS metrics file. This method requires project descriptor and returns the URL to the metrics zip file.
- 4) `downloadProjectMetricsMTOM()`
Downloads the XLS metrics file. This method requires project descriptor and returns the MTOM DataHandler. The data handler contains the metrics zip file.
- 5) `obtainProjectStatistics()`
Obtains up to date user statistics for all jobs for the whole project for each target language or one specified target language. This method requires project descriptor and optional target language in the additional method options.

15. Generate, Check the Status and Download Files

1. `generateJobFile()`
Creates a file with a given type (`GENERATED_FILE_TYPE`) for specified jobs. This method requires a list of jobs descriptors and returns files descriptors which should be used in the methods described in points 2, 3 and 4.
2. `checkJobFileCompletion()`
Checks if the given files are generated. This method requires a list of files descriptors and returns status for each file (`XTM_JOB_FILE_COMPLETION_STATUS`).
3. `downloadJobFileURL()`
Downloads all files which are generate. This method requires a list of files descriptors and returns URLs to files.
4. `downloadJobFileMTOM()`
Downloads all files which are generated. This method requires a list of files descriptors and returns MTOM DataHandler('s). Data handler contains zip file.

`GENERATED_FILE_TYPE` - possible file types that can be generated

```
/** Target file. */  
TARGET,  
  
/** XLIFF file. */  
XLIFF,  
  
/** QA Report. */  
QA_REPORT;
```

`XTM_JOB_FILE_COMPLETION_STATUS` - possible job file completion statuses

```
/** File is still generating. */  
IN_PROGRESS,  
  
/** File has been generated. */  
FINISHED,  
  
/** File could not be generated. */  
ERROR;
```

16. Download Files

1. `downloadProjectMTOM()`
Downloads all jobs from the given project which are finished. This method requires project descriptor and returns MTOM DataHandler('s). Data handler contains zip file.
2. `downloadProjectURL()`
Downloads all jobs from the given project which are finished. This method requires project descriptor and returns URLs to files.
3. `downloadJobMTOM()`
Downloads all the given jobs which are finished. This method requires a list of jobs descriptors and returns MTOM DataHandler('s). Data handler contains zip file.
4. `downloadJobURL()`
Downloads all the given jobs which are finished. This method requires a list of jobs descriptors and returns URLs to files.

17. Obtain Additional Data

1. `obtainJobExtraInfo()`
Obtains additional identifiers for the given list of jobs. This method requires a list of jobs descriptors and returns the list of `XTMJobExtraInfoResponseAPI` objects.
2. `obtainProjectExtraInfo()`
Obtain additional identifiers for all jobs in the given project. This method requires project descriptor and returns the list of `XTMJobExtraInfoResponseAPI` objects.
3. `getXTMInfo()`
Obtain the base information about XTM. Returns the `XTMInfoResponseAPI` object.
4. `getSupportedFilesInfo()`
Obtains the list of supported and not supported files' extensions. For some not supported files there is a list of alternative extensions which can be used instead of the given file, for example: DOC and DOCX can be used instead of DOCM.

`XTMJobExtraInfoResponseAPI` - additional identifiers

```

/** Job descriptor. */
jobDescriptor,

/** ID of the XLIFF file. */
xliffFileId,

/** Project descriptor. */
projectDescriptor,

/** ID of the target language. */
targetLanguageId,

/** Name of the XLIFF file. */
xliffFileName,

/** ID of the source file. */
originalFileId,

/** Name of the original file. */
originalFileName;

```

`XTMInfoResponseAPI` – base information about XTM

```

/** Company name. */
companyName,

/** URL to the website. */
website,

/** Path to logo image. */
logo,

/** XTM Version. */
version;

```

18. Upload an XLIFF File

1. `uploadXliffURL()`
Uploads a list of XLIFF files for the given job and step descriptors. Autopopulate in additional options will set default values. This method returns the files descriptors which should be used in the method described in point 3.
2. `uploadXliffMTOM()`
Uploads list of XLIFF files for the given job and step descriptors using MTOM data handlers. Autopopulate in additional options will set default values. This method returns the files descriptors which should be used in the method described in point 3.
3. `checkUploadXliffCompletion()`
Checks if XLIFF files are uploaded. This method requires a list of files descriptors and returns the status for each XLIFF file (`XTM_UPLOAD_XLIFF_COMPLETION_STATUS`).

`XTM_UPLOAD_XLIFF_COMPLETION_STATUS` - possible XLIFF completion statuses

```
/** XLIFF is still uploading. */  
IN_PROGRESS,  
  
/** XLIFF upload is completed successfully. */  
FINISHED,  
  
/** XLIFF upload could not be completed. */  
ERROR,  
  
/** The given XLIFF does not exist. */  
NOT_EXIST;
```

19. Open XTM modules

1. getEditorURL()

Returns URL to the Editor for the given XLIFF file.

EDITOR_ROLE_API - possible editor roles

```
/** Translator role. */  
TRANSLATOR,
```

```
/** Corrector role. */  
CORRECTOR,
```

```
/** Reviewer role. */  
REVIEWER;
```

EDITOR_TERMINOLOGY_RIGHTS_API - possible editor terminology rights

```
/** Add right. */  
ADD,
```

```
/** Modify right. */  
MODIFY,
```

```
/** View right. */  
VIEW,
```

```
/** Import right. */  
IMPORT,
```

```
/** Export right. */  
EXPORT;
```

2. obtainPMProjectEditorLink()

Returns the URL to the PM Project Editor for the given project. It requires project descriptor and optional settings which can be specified in the PMProjectEditorOptions object. Options allow to set visibility for the following tabs: generals, metrics, statistics, workflow, files, estimates. By default all tabs will be visible.

3. obtainPMTranslationPropsLink()

Returns the URL to the PM Translation Properties page. This configuration can be opened for the whole system or the given customer. Additional options in PMTranslationPropsOptions object allow to set visibility for the following parts: translation, application options, segment status, machine translation.

The following data should be filled:

- configuration level
- customer descriptor – only when configuration level is set to Customer

ConfigurationLevel – possible configuration levels

```
/** Configuration for Client. */  
CLIENT,
```

```
/** Configuration for Customer. */  
CUSTOMER;
```

20. PHP Examples

20.1. Web Service Framework for PHP Installation Guide

Tested on Windows 7(64), PHP-5.3.2-Win32-VC9-x86, Apache 2.2.15(Win32).

- Install Apache, PHP (set XSL extension)
- Copy wsf.dll to the <PHP installation directory>/ext directory.
- Add wso2-wsf-php-bin-x.y.z-win32\wsf_c\lib directory to the PATH environment variable.
- Add the following entries to your php.ini file which is in your PHP installation location.

```
[wsf]
wsf.home="<path_to_extract_folder>\wsf_c"
wsf.log_path="<path to extract_folder>\wsf_c\logs"
wsf.log_level=3
extension=wsf.dll
```

Make sure that extension_dir entry is set to your extensions directory and php_xsl extension is enabled

```
extension_dir = ".\ext"
```

```
extension = php_xsl.dll
```

Uncomment the include_path and edit it to point to the location for scripts folder as
include_path = ".;<path to the WSF/PHP scripts folder>"

- Copy php5apache2_2.dll to Apache2/modules directory and add the following entries in httpd.conf file.

```
LoadModule php5_module modules/php5apache2_2.dll
PHPIniDir "< your php.ini file location> "
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

20.2. createProjectURL

```

//////////////////////////////////// BEGIN client configuration
$wsdl = "http://SERVER/project-manager-gui/services/v2/XTMCustomerMTOMWebService?wsdl";
$sourcefile = 'http://www.w3schools.com/xml/simple.xml'.urldecode($_GET['xml']);

// variables:
$loginAPI = array('userId' => USER_ID, 'password' => 'PASSWORD', 'client' => 'COMPANY_NAME');
$xmlCustomerDescriptorAPI = array('id' => 1265); // existing customer ID

$sourceLanguage = 'en_GB'; // source language
$targetLanguages = array('ru_RU','pl_PL'); // list of target languages

//variables set after project creation
$jobsDescriptors = array();
$projectDescriptor = null;

$client = new WSClient(array('wsdl' => $wsdl));

$proxy = $client->getProxy();

//////////////////////////////////// END client configuration

printf("<i>createProjectURL</i><br />");

$xmlFileURLAPI = array('fileURL' => $sourcefile);
$xmlProjectURLAPI = array('translationFiles' => array($xmlFileURLAPI),
                        'sourceLanguage' => $sourceLanguage,
                        'targetLanguages' => $targetLanguages,
                        'customer' => $xmlCustomerDescriptorAPI,
                        'name' => 'PHP API project_1');

$xmlCreateProjectURLOptionsAPI = array('autopopulate' => true);

//invoke
$input = array('loginAPI' => $loginAPI, 'project' => $xmlProjectURLAPI, 'options' =>
$xmlCreateProjectURLOptionsAPI);
$res = $proxy->createProjectURL($input);

//checking result
$xmlCreateProjectURLResponseAPI = $res['return'];
$xmlProjectResponseAPI = $xmlCreateProjectURLResponseAPI['project'];

printf("Project name: ' . $xmlProjectResponseAPI['name'] . '<br />");
printf("Project ID: ' . $xmlProjectResponseAPI['projectDescriptor']['id'] . '<br /><br />");

printf("Jobs: <br/>");
foreach ($xmlProjectResponseAPI['jobs'] as $xmlJobResponseAPI)
{
    printf("- Job ID: ' . $xmlJobResponseAPI['jobDescriptor']['id'] . ', ');
    printf("File name: ' . $xmlJobResponseAPI['fileName'] . ', ');
    printf("Target language: ' . $xmlJobResponseAPI['targetLanguage'] . '<br />");
    array_push($jobsDescriptors, $xmlJobResponseAPI['jobDescriptor']);
}

$projectDescriptor = $xmlProjectResponseAPI['projectDescriptor'];

```

20.3. checkProjectAnalysisCompletion

```
//include configuration section from createProjectURL method (20.2)

printf("<br /><br /><i>checkProjectAnalysisCompletion</i><br />");

while (true)
{
    //invoke
    $input = array('loginAPI' => $loginAPI, 'project' => $projectDescriptor);
    $res = $proxy->checkProjectAnalysisCompletion($input);

    $xtmCheckProjectAnalysisCompletionResponseAPI = $res['return'];
    $xtmProjectCompletionStatus = $xtmCheckProjectAnalysisCompletionResponseAPI['project']['status'];

    if ('FINISHED' == $xtmProjectCompletionStatus)
    {
        printf('Project analysis has been finished, ID: ' .
$xtmCheckProjectAnalysisCompletionResponseAPI['project']['projectDescriptor']['id'] . "<br />");
        printf('Jobs: <br />');

        foreach($xtmCheckProjectAnalysisCompletionResponseAPI['project']['jobs'] as
$jobResponse)
        {
            printf("- Job ID: " . $jobResponse['jobDescriptor']['id'] . ", status: " .
$jobResponse['status'] . "<br />");
        }
        break;
    }
    else
    {
        printf('Checking...<br>');
    }

    sleep(2);
}
}
```

21. JAVA Examples

Java examples are available from XTM International in a ZIP file which can be easily imported to your Eclipse environment as a new project. The file contains:

- Examples
 - Create project using URLs method
 - Check analysis completion

- All required libraries to run prepared code

If you are interested in the Java examples and would like to obtain a copy of them please contact sales@xtm-intl.com



XML International Ltd, PO Box 2167, Gerrards Cross, SL9 8XF, UK
el.: +44 (0)1753 480467 Fax: +44 (0)1753 480465 email: sales@xtm-intl.com <http://www.xtm-intl.com>