

Smart Connectors - Technical Guide

License Agreement

The article presents the features available in the Premium plan of XTRF Language Business Platform. Please keep in mind that your access to the described options might be limited due to your license agreement. If you would like to change your plan of XTRF Language Business Platform and gain access to the additional features, contact your [XTRF Customer Success Manager](#).

Target audience: Developers

Preconditions: JavaScript knowledge

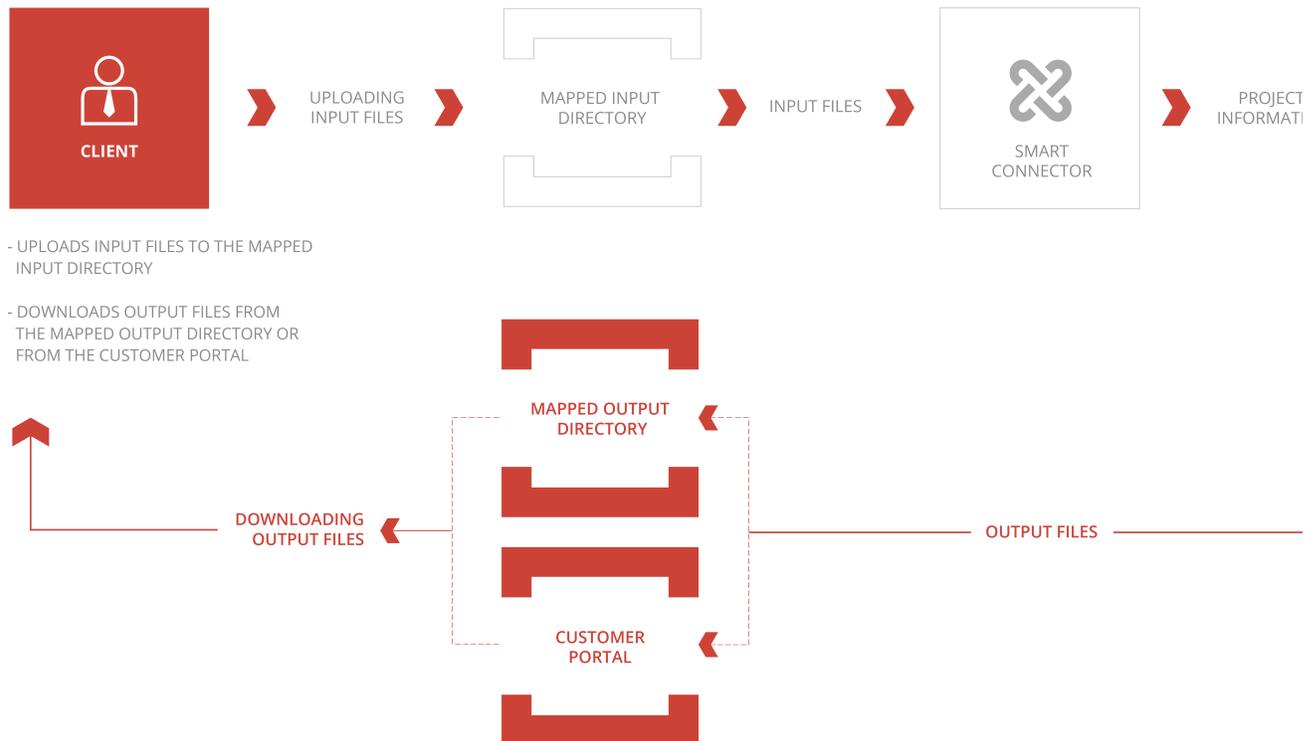
- Overview
- External File Sharing Service
- Technical Configuration
 - Basic Smart Connector
 - Advanced Smart Connector
 - Smart Connector with a Descriptor
 - Available Parameters
- Available Features
 - Uploading Files to Smart Connector
 - Supported Files
 - File Names
 - Managing Files
 - E-mail Notifications
 - Scheduling
- Testing Smart Connectors Configuration

▲
Back to
Administrator's
Guides



Overview

Smart Connectors allow for quick predefined project creation, however, the functionality may be very flexible in hands of users with advanced knowledge of JavaScript. If you work with a client who regularly orders very similar translation projects, you can set up the Smart Connector for them. Such Smart Connector saves Project Manager's time. A well-prepared Smart Connector, creates a project automatically when it detects files at the specified time in the monitored directory. When the project is finished, the target file is uploaded to the directory that can be specified in the Smart Connector's settings, however, the directory needs to be provided in the Smart Connector's robot code. The client can download it from there if they are given access to that location. In this article you can find all information necessary to set up a Smart Connector for your client in the configuration of XTRF Platform. Keep in mind that each Smart Connector requires different mapped location and each client needs to have separate Smart Connector assigned to avoid any issues.



External File Sharing Service

Smart Connector operates on the files uploaded to the mapped location. The location can be defined on your local machine, on an FTP server, SFTP server or even e-mail. Choose one file sharing platform or prepare an e-mail address and provide its directory which will be mapped in Smart Connector in the Scan for Input Files section. For example, if you want to map an FTP server, provide the proper pieces of information in the available fields in the Smart Connector's settings.

Your client does not need to configure anything in terms of using external file sharing service integrated with Smart Connectors. They can choose one of the following options to deliver the source files to their Project Manager:

- They can send the files to the Project Manager, who will upload the files to the mapped folder
- You can give them access to the mapped folder, therefore they can upload files themselves.
- If you prepared the Smart Connector in the way that it scans an e-mail address, your client can send e-mails with attachments and the Smart Connector will start a project based on the files attached in the message. If your proficiency in JavaScript is advanced enough, you can even set up a Smart Connector which can access files through the link provided in the e-mail.

When the files are uploaded to the mapped folder and the Smart Connector is correctly configured, the project can be created.

Technical Configuration

In this section you can learn how different kinds of Smart Connectors can be configured. Depending on how well you can handle JavaScript, you can configure more or less advanced Smart Connectors. However, you do not need any knowledge of JavaScript to set up basic Smart Connector.

Each Smart Connector needs to be configured in XTRF Platform. You can do that using a wizard available in the Home Portal. In

the picture below you can see the configuration of the Smart Connector prepared in the configuration wizard just before saving. The code snippets provided further in the article can be typed in the Connector Definition code field. To enable code field you only need to select the Robot Code radio button. Bear in mind that the Name and Client fields need to be filled before saving the Smart Connector or switching to other tabs.

- Add Smart Connector

General Settings	Schedule	E-mail Notifications	Logs
Name	<input type="text" value="Smart Connector For Company 1"/> *		
Active	<input checked="" type="checkbox"/>		
Client	<input type="text" value="Andrew Wilson"/> *		
Client Price Profile	Euro [€] - Euro [€]		
Files Aggregation	<input type="text" value="Process every top level file separately"/>		
Scan for Input Files	<div style="margin-bottom: 5px;"> i Your files are being moved from the Input and attached to the newly added projects. </div> <div style="margin-bottom: 5px;"> <input checked="" type="radio"/> XTRF directory <input type="radio"/> FTP <input type="radio"/> SFTP <input type="radio"/> E-mail </div> <div style="margin-bottom: 5px;"> Directory <input type="text" value="/home/jboss/xtrf/smartConnector"/> * </div>		
After Successfully Processing, Store Files in	<input type="radio"/> XTRF directory <input type="radio"/> On FTP <input type="radio"/> On SFTP <input checked="" type="radio"/> Do not store		
In Case of Errors, Store Files in	<input type="radio"/> XTRF directory <input type="radio"/> On FTP <input type="radio"/> On SFTP <input checked="" type="radio"/> Do not store		
Person Responsible	<input type="text" value="Greg Smith"/>		
Connector Definition	<div style="margin-bottom: 5px;"> <input checked="" type="radio"/> Wizard <input type="radio"/> Robot Code </div> <div style="margin-bottom: 5px;"> Source Language <input type="text" value="German [DE]"/> </div> <div style="margin-bottom: 5px;"> Target Language <input type="text" value="English [EN]"/> </div> <div style="margin-bottom: 5px;"> Service <input type="text" value="Translation Premium"/> </div> <div style="margin-bottom: 5px;"> Specialization <input type="text" value="IT"/> </div>		

Basic Smart Connector

Basic Smart Connector can be set up in the configuration wizard, therefore you do not need to know how to use JavaScript. The configuration wizard allows for setting up only the most simple Smart Connector. This kind of Smart Connector is prepared to create projects in fixed language configuration as well as service and specialization. It means that it can be applied for the client who orders multiple projects in very similar configurations.

Simple Smart Connector - Code Snippet

```
function robot(input) {  
  return {  
    project: {  
      sourceLanguage:  
utils.repository.languages.bySymbol('EN'),  
      targetLanguages: [  
utils.repository.languages.bySymbol('DE') ],  
      service:  
utils.repository.services.byName('Translation'),  
      specialization:  
utils.repository.specializations.byName('General'),  
      files: input.files  
    }  
  };  
}
```

The project created by the Smart Connector example will involve translation from English to German and the set workflow will be appropriate for the Translation service. The specialization is set to General.

The code snippet provided above was created using the configuration wizard. Similar codes are created every time the configuration wizard is used.

Advanced Smart Connector

The advanced Smart Connector allows you to set more parameters for the project, however, some knowledge of JavaScript is required.

The Smart Connector example provided below creates a project named User Manual Translation. The source language is German and the target languages are Polish, PL, and English, EN. The service involved in the project is set to Translation Premium , since its retrieved from the XTRF's database, the workflow applied in the project will be as specified in the Premium Translation service. The specialization of the translation is set to the system's default and the category is Automatic-Project. Current date is set as the start date for the project and the deadline is set to a week after the start date. The directory of the translated files is set to this/its/my/relative/path on an FTP server. Notes that will be displayed in the project's Instructions tab are as follows:

- Important notes - for the Notes section,
- Important notes from client - for the Special Instructions from Client section,
- Important internal instructions - for the Internal Special Instructions section.

The code snippet needs to be entered in the Connector Definition field in the Smart Connector configuration. In the Create Smart Connector display select the Robot Code radio button to enable the code field.

The Smart Connector in the example has fixed parameters, it means that projects created by this Smart Connector will have the same configurations. You can use it for the client who orders multiple translation projects of the same type.

Advanced Smart Connector

- Code Snippet

```
function robot(input) {
  var now = new Date().getTime();
  return {
    project: {
      name: 'User Manual Translation',
      sourceLanguage:
utils.repository.languages.bySymbol('DE'),
      targetLanguages: [
utils.repository.languages.bySymbol('PL'),
utils.repository.languages.bySymbol('EN') ],
      service:
utils.repository.services.byName('Translation Premium'),
      specialization:
utils.repository.specializations.getDefault(),
      categories: [
utils.repository.categories.byName("Automatic-Project") ],
      notes: "Important notes",
      instructionsFromCustomer:
"Important instructions from client",
      internalInstructions: "Important internal instructions",
      startDate: new Date(now),
      deadline: new Date(now + 7*24*60*60*1000),
      files: input.files,
      delivery: {
        url:
'ftp://localhost:21/this/is/my/relative/path'
      },
    },
  };
}
```

Smart Connector with a Descriptor

Both simple and advanced Smart Connectors provided as examples above contain some fixed settings. Therefore, every time a project is created using one of those Smart Connectors, the language combinations or any other parameters specified in the code of those Smart Connectors will be applied. A Smart Connector with a descriptor can be used for multiple projects with completely different parameters, which can be specified in the descriptor file. A descriptor file with project specifications needs to be prepared

and uploaded with the file that needs to be translated for every new project created in this type of Smart Connector. An appropriate Smart Connector needs to be prepared for interpreting of the descriptor file.

The code snippet should be entered in the Connector Definition section. In the Create Smart Connector display select the Robot Code radio button to enable the code field.

The Smart Connector in the example provided below is prepared to retrieve the information from the descriptor file. Name of the project, source and target languages, service and notes will be retrieved from the descriptor.

The descriptor needs to be saved as a `.json` file and added together with the files that are to be the subject of the project.

Smart Connector with a Descriptor - Code Snippet

```
function robot(input) {
  var descriptor;
  var projectFiles = [];
  var now = new Date().getTime();
  input.files.map(function(file){
    if (file.relativeDirWithName ==
'descriptor.json') {
      descriptor = file.loadAsJSON();
    } else {
      projectFiles.push(file);
    }
  });
  if ( ! descriptor) {
    throw 'no descriptor.json found in
input files';
  }
  return {
    project: {
      name: descriptor.name,
      sourceLanguage:
utils.repository.languages.bySymbo
l(descriptor.sourceLanguage),
      targetLanguages:
descriptor.targetLanguages.map(fu
nction(targetLanguage) {
        return
utils.repository.languages.bySymbo
l(targetLanguage);
      }
    ),
      service:
utils.repository.services.byName(de
scriptor.service),
      specialization:
utils.repository.specializations.getD
efault(),
      notes: descriptor.notes,
      startDate: new Date(now),
      deadline: new Date(now +
7*24*60*60*1000),
      files: projectFiles
    }
  };
}
```

Descriptor - Code Snippet

```
{  
  "name": "Translation Project",  
  "sourceLanguage": "DE",  
  "targetLanguages": ["PL", "EN", "FR"],  
  "service": "Translation Plus",  
  "notes": "Important notes"  
}
```

You can see an example of a descriptor file. The data provided for the Smart Connector by the descriptor are as follows:

- name
- source language
- target languages
- service
- notes

The Smart Connector retrieves the information from the `descriptor.json` file and creates project based on that information. The new project will be named Translation Project, the source language will be German, DE, and there will be three translation tasks: into Polish, PL, English, EN, and French, FR. The project's workflow will be appropriate for the Translation Plus service uploaded to the XTRF's database and in the project's Instructions tab the information Important notes will be displayed in the Notes section.

Keep in mind that the data provided in the descriptor file need to be compatible with the data uploaded to your XTRF Platform.

Available Parameters

In the table below you can find the list of all parameters available for creating an advanced Smart Connector and the descriptor file. Parameters necessary to create the project are marked as such in their description, all the parameters that are not marked as necessary are optional and are not required to create a project.

While configuring Robot Code, the system may provide you with suggestions for possible operations. To get suggestions press CTRL and SPACE on your keyboard while typing in the code field. A list with possible operations appears.

Parameter	Description of the Parameter
name	This parameter sets the name for the project created by the Smart Connector. If the parameter is not added to the Smart Connector's code, the name of the project will be empty.

sourceLanguage	<p>This parameter specifies the source language in the project. The source language needs to be retrieved from the repository, therefore, the appropriate language needs to be uploaded to the XTRF's database. This parameter is necessary to create a project.</p> <div data-bbox="493 306 716 600" style="border: 1px solid black; padding: 5px;"><p style="text-align: center;">Example</p><pre>sourceLanguage: Repository.lan guages.bySym bol('EN-US')</pre></div>
targetLanguages	<p>This parameter specifies the target languages in the project. The target languages need to be retrieved from the repository, therefore, appropriate languages need to be uploaded to the XTRF's database. This parameter is necessary to create a project.</p> <div data-bbox="493 831 716 1209" style="border: 1px solid black; padding: 5px;"><p style="text-align: center;">Example</p><pre>targetLanguag es: [repository.lan guages.bySym bol('DE'), repository.lan guages.bySymb ol('FR')]</pre></div>
service	<p>This parameter specifies what kind of service is provided in the project. The Smart Connector applies the workflow to the project according to this parameter. The services and their corresponding workflows need to be retrieved from the repository, it means that both services and workflows need to be configured and uploaded to the XTRF's database. This parameter is necessary to create the project.</p> <div data-bbox="493 1440 716 1734" style="border: 1px solid black; padding: 5px;"><p style="text-align: center;">Example</p><pre>service: repository.serv ices.byName(' premium translation')</pre></div>

<p>specialization</p>	<p>This parameter specifies the specialization of the translation project. The specialization needs to be retrieved from the repository, therefore, the appropriate specialization needs to be uploaded to the XTRF's database.</p> <div data-bbox="493 302 717 571" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Example</p> <pre>specialization: repository.specializations.getDefault()</pre> </div>
<p>categories</p>	<p>This parameter specifies the category of the project. The category needs to be retrieved from the repository, therefore, the appropriate categories need to be uploaded to the XTRF's database.</p> <div data-bbox="493 716 717 1016" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Example</p> <pre>categories: [repository.categories.byName("automatic-project")]</pre> </div>
<p>notes</p>	<p>This parameter allows to add notes to the project. You can add the content of the note in the quotation marks in this parameter. The notes will be displayed in the project's Instructions tab in the Notes section.</p>
<p>instructionsFromCustomer</p>	<p>This parameter allows to add instructions from the client. The content of the instructions can be added in the quotation marks in this parameter. The instructions will be displayed in the project's Instructions tab in the Special Instructions from Client section.</p>
<p>internalInstructions</p>	<p>This parameter allows to add special internal instructions to the project. The content of the instructions can be added in the quotation marks in this parameter. The instructions will be displayed in the project's Instructions tab in the Internal Special Instructions section.</p>

startDate	<p>You can add the start date of the project using this parameter. If you provide a specific date, the date will be fixed for this Smart Connector. It means that every time this Smart Connector is used to create different projects, the start date will be always the same as specified in this parameter. You can also set the current date as the start date or add a period of time in milliseconds to the current date, if you wish the start date to be in future, for example $7*24*60*60*1000$ for seven days.</p> <div data-bbox="493 407 717 798" style="border: 1px solid #ccc; padding: 10px;"><p style="text-align: center; background-color: #f0f0f0; margin: -10px -10px 10px -10px;">Example</p><pre>var now = new Date() .getTime(); ... startDate: new Date(now + 7*24*60*60*1 000)</pre></div>
deadline	<p>You can specify the deadline for the project. You can either provide a specific date, which will then be fixed for this Smart Connector, or you can set the current date or the start date and add a period of time in milliseconds, if you want the deadline to be somewhere in future. For example you can add $7*24*60*60*1000$ milliseconds to add 7 days.</p>
files	<p>This parameter specifies the list of the source files for the project.</p>
customFields	<p>The parameter specifies values of your custom fields for the project. Use key or label to identify custom field.</p> <div data-bbox="493 1121 717 1822" style="border: 1px solid #ccc; padding: 10px;"><p style="text-align: center; background-color: #f0f0f0; margin: -10px -10px 10px -10px;">Example</p><pre>customFields: { myTextField: 'text', myCheckbox: true, mySelection: 'option1', myNumber: 2.56, myMultipleSel ection: ['option1','opti on2'], myDate: new Date (2000000000) }</pre></div>

delivery	<p>You can specify the directory path for the translated files. The translated files will be uploaded to this directory when the project is finished.</p> <div data-bbox="493 275 717 606" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"><p style="text-align: center; background-color: #f0f0f0; margin: 0;">Example</p><pre>delivery: { url: 'ftp://localhost :21/this/is/my /relative/path' }</pre></div>
-----------------	---

Available Features

Uploading Files to Smart Connector

Once the folder for uploading files is mapped or an e-mail address is specified, you or your client can copy the files to the folder or send the e-mail with attached files. Multiple files can be uploaded to the mapped folder or attached to an e-mail. Depending on the Smart Connector's settings, it can either create separate projects for each file or one project for all the uploaded files.

Supported Files

Smart Connectors support any kind of file. The file restrictions depend on what happens with the file after the project has been created. For example, if the file is to be uploaded to memoQ translation software, then it should be compatible with memoQ specifications. Smart Connectors do not impose any restrictions on the provided files.

File Names

There are no file name restrictions.

Managing Files

Once a file is uploaded to the folder mapped or an e-mail with attached files is sent to the address specified in the Scan for Input Files section and the Smart Connector is started, the Smart Connector creates a project. The files are available for the vendors assigned to the project as if the project was created manually. Depending on the configuration of the Smart Connector, once the project has been started, the Smart Connector can create a folder with the project's ID number in the directory specified for the processed files and can store the input files there or remove them. If there are any problems with the uploaded file, which would prevent creating a project, the Smart Connector can create a subfolder with the exact date in the name in the directory specified for the In Case of Errors, Store Files in section or it can remove them.

You need to provide the entire paths in the directories of specific sections.

You can add output path for the target files in the Robot Code of the Smart Connector. If the project includes more than one translation task, the target files are stored in folders with a proper language symbol as a folder name.

Multiple files can be added at the same time as a **.ZIP** archive. The Smart Connector recognizes the **.ZIP** archive, creates a subdirectory with the same name as the **.ZIP** archive and extracts the files in the subdirectory.

If the name of the file or **.ZIP** archive contains characters unsupported by Microsoft Windows operating system, such as: *** , : , ? , < , > , or |** , the Smart Connector replaces those characters with **_**

Once the project has started, additional files cannot be uploaded to the project using Smart Connector functionality.

E-mail Notifications

You can select when and to whom the e-mail notifications are sent.

The notifications can be sent when:

- The project is created. The notification is sent to person responsible for the project created with the Smart Connector.
- An error occurs. When error occurs and project cannot be created the notification is sent to the person responsible for the Smart Connector and the project.
- Project is requested with Smart Connector. The notification is sent to the client.

Scheduling

You can specify when the Smart Connector attempts to create a project. The project can be started:

- As soon as the Smart Connector detects files in the mapped folder,
- At a scheduled time. You can specify when the Smart Connector should scan the mapped folder and attempt to run a project:
 - Once per day,
 - Repeatedly per day,
 - On selected days of the week,
 - Once a month,
 - You can also provide a custom schedule, however it needs to be provided as Cron Expression.

Testing Smart Connectors Configuration

To make sure that the Smart Connector will work properly and the folder in Base Directory is mapped correctly, you can perform the following steps:

1. Sign in to the Home Portal of XTRF Platform.
2. Click the  icon.
3. In the accordion configuration menu select the Integration and then select Smart Connectors.
4. Click the Add button in the upper side of the Smart Connectors list.
5. Prepare a Smart Connector: Upload a file to the folder specified in Input section.
 - a. Provide the name for the Smart Connector in the Name field.
 - b. Select the client to whom the Smart Connector is dedicated. The system provides the suggestions as you start typing the name of your client.
 - c. Select the client's Price List.
 - d. In the Files Aggregation drop-down menu select the way the Smart Connector will handle the files uploaded to the mapped folder.
 - e. Provide the directory of the folder or the access data for the e-mail which will be scanned for input files.
 - f. Select what happens with processed files and provide directory for them if necessary.
 - g. Select what happens with the files if any error occurs and provide directory if necessary.
 - h. Select the Person Responsible for the Smart Connector
 - i. Select the Wizard radio button in the Connector Definition field if you want to prepare a simple Smart Connector or Robot Code radio button if you want to prepare an advanced Smart Connector.
 - j. Select the specifications of the project from the drop down menus if you selected Wizard radio button or provide the piece of JavaScript code if you selected Robot Code radio button.
 - k. Click the Save button.
6. Click the Test button. Test Smart Connector pop-up window containing logs appear. If the directory in Input section is mapped correctly, the file is processed and a pop-up window informing about the potential project appears. In the pop-up window you can choose to create a project or close the pop-up window.

Please refer also to the related articles:

- Basic Definition:
 - [Smart Connectors Definition](#)
- XTRF Modules Context-Sensitive Help
 - [Browse Smart Connectors - Integration](#)
 - [Smart Connectors - Integration](#)
 - [General Settings - Smart Connectors - Integration](#)
 - [Schedule - Smart Connectors - Integration](#)
 - [Notifications - Smart Connectors - Integration](#)
 - [Logs - Smart Connectors - Integration](#)
- User's Guides:
 - [Configuring Smart Connector](#)

The license could not be verified: License Certificate has expired!