

Delivery methods (a.k.a. Actions)

How a job is actually done in Plint is defined by its Delivery method (also called an Action). Additional methods are continuously added.

- [Report](#)
- [Online editing](#)
- [Online review](#)
- [Media upload](#)
- [File upload](#)
- [Review/QC using MQM tools](#)
- [Auto create subtitles](#)
- [Auto FFMPEG render](#)
- [Auto delivery](#)
- [Auto upload](#)
- [ASR](#)
- [Automated jobs and job parenting](#)
- [KNP / Term bases](#)

Report

The simplest delivery method allows a user to click a button and report the job as Done. This will change the status of the job and look for child jobs. If any are found, job owners are notified that their work can now begin. Setting the job as Done also removes it from the user's Dashboard.

Report jobs are good for offline and custom work that is a required part of a workflow but do not correspond to another delivery method.

A Report job on the user Dashboard.

A Sunny Day in Projectville

ID: 8

☐	Status	ID	Title	Part	Language	Delivery date	
☐	In process	338	Prepare and report		N/A	N/A	Report as Done

Online editing

Online editing jobs allow any user to originate or translate in Plint using their browser. The common logic around online editing is explained [here](#). Plint Subtitler is covered in detail [here](#).

Online review

Akin to online editing, Online review enables files in standard formats (doc, video, pdf ...) to be reviewed and refined in a simple workflow:

- User A: Upload file
- User B: Review in browser
 - Approve (done)
 - Not approve
 - Parent job reopened for new upload by User A
- Repeat until perfect

As with online editing, you can add a general comment on status change and a history of versions is maintained.

Dashboard / Projects / A Sunny Day in Projectville / Marketing materials / Review

or1_first_draft.pdf

The first draft

The subheader

This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document.

This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document. This PDF is the first draft of a very important document.

[Download](#) [View on new tab](#) [Supported files](#)

Job info

Project: A Sunny Day in Projectville

Job: [Marketing materials #339](#)

Status: In process

Part: 2 - no

[Upload new file](#) [Change status](#)

Media upload

This delivery method serves a very specific purpose: to supply the system with video (a "proxy") which will be automatically loaded into the online editors. Using this method will automatically set the right **File mode** and link it to the same **node** as the job. Only a single file can be uploaded.

The file will also be analysed after upload. The extracted metadata, such as resolution and framerate, can be viewed by clicking the "Information" icon for the file in the Filezone.

The running time may also be automatically entered on part level for the project. Please note that this is only done if the **site** has this global setting turned on and the **project unit** is set to minutes.

Files can also be automatically converted into an editor-friendly format using the built-in transcoder.

More [here](#).

Media upload

Name

bars_and_timecode_24fps.mp4

Size

1.6 MB

File description

Comment to project manager

☐ Make visible on files tab

This will allow users with matching jobs to download the file.

Convert after upload?

Upload without conversion ▼



Timecode start time

HH:MM:SS:FF

Max upload size: 25000MB

Upload

Cancel

File upload

Unlike Media upload, File upload jobs support multiple files. This delivery method simply "delivers" the uploaded files into the project [Filezone](#) (setting them to File mode "Standard" by default). After upload, the user can add description and other information for any or all files before changing status.

"File upload - advanced" works the same way but allows the user to specify more information/options per file.

The information that can be specified for each file includes:

- **File description**
 - can be any text
- **File categories** are edited under *Tools & Data*
 - If one or more categories are mandatory, the File upload job cannot be completed until this category has been linked to a file
 - Which categories are available for selection for a certain upload job is determined on job level
 - This is done under the "File settings" tab

Adding file information before status change

Job: 340 | Upload multiple files | Part: 3 | File upload

Max upload size: 25000MB

Drop files here to upload

Name	Ext	Size	Uploaded to	File mode	Upload date	Actions
Glossary	txt	1 B	Uploaded to this job by you	Standard	2020-03-10 13:58:42	  

File category

Time offset

HH:MM:SS:FF

Framerate

25

Version

BR21

File description

BR version awaiting final approval from TR\HQ

Metadata	txt	1 B	Uploaded to this job by you	Standard	2020-03-10 13:58:42	  
----------	-----	-----	-----------------------------	----------	---------------------	---

File category

Review/QC using MQM tools

[Plint Subtiter](#) supports a review mode where each change/error is logged according to a standardised set of issue types. Using these tools involves a combination of tools and settings rather than a single Delivery method:

MQM (Multidimensional Quality Metrics) is a framework for objective assessment of translation quality, using standardized error categories and severity levels to calculate a quality score. Depending on the demands, categories can be adapted/omitted and the threshold levels for acceptance adjusted.

Adjusting and understanding the MQM settings in Plint

Under *Tools & Data / MQM Settings*, a site admin specifies how the various error categories will affect the report. Each reported issue will add a number of points to the accumulated error score. This number depends on what value is entered here for the selected *Category* and *Severity* (e.g. "Minor"), multiplied with the *Weight*. In the below example, a Major Mistranslation would result in 120 points.

The "Max allowed errors" setting determines if a job will Pass or Fail, based on the following calculation: is the combined score per **1000 words** higher or lower than this threshold? This calculation is done when the report is completed and summarised (see below).

It is highly recommended that you make any such changes early on, to ensure that reports aggregated over time can be compared in a meaningful way.

General settings

Max allowed errors

300

Err/1k

Error categories

ACCURACY							
The translation does not accurately convey the meaning of the source text, unless specific instructions allow for this.	Score				Repeat	Weight	Active
	Neutral	Minor	Major	Critical			
Mistranslation	0	10	30	300	0	4	☒
Mishear	0	10	30	300	0	4	☒
Addition	0	10	30	300	0	3	☒
Omission	0	10	30	300	0	3	☒
Untranslated text	0	10	30	300	0	3	☒

Plint Subtitler in MQM mode

The actual review is done in **Plint Subtitler**, in a dedicated MQM mode. [How to use this mode is explained here](#). The mode is activated on the Editor settings tab:

General

Editor settings

Casting

Job Fee

Job history

Messages

Notes to self

File settings

Online editing/review settings

 Proofing-type

No

 Text translation mode

No

 MQM Review Mode

No



An MQM Report job (optional)

This job consists of viewing the report after the quality control has been completed. All reports are stored and linked to the users (Reporter / Translator) and can be accessed as needed for the users in question. However, in order to quickly assess the result of a test, you can create a job and assign it to a project manager, vendor manager or similar.

You create such a job with the Delivery method "MQM Report".

The job itself will appear on the user's Dashboard with an icon leading to the report (red or green, depending on Pass/Fail status). There is also a button to report the job as Done and remove it from the Dashboard. Please note that this will not remove the report from the system:

Evaluation demo - Fjällbacka

New order

ID: 22

📅 N/A

Jobs: 28

PM: Anders L

🔍

✎

✎

☰

📄

👥

📄

+

💬

🕒

📥

📺

🔄

📧

📄

67%

33%

☐	Status	ID	Title	Part	Language	Delivery date	
☐	In process	597	Evaluation Report	1	Swedish	N/A	Report as Done 📄
☐	In process	600	Evaluation Report	1	English	N/A	Report as Done ✖

Reading the report

The report will summarise the result by showing the number of individual errors and their categories and severity, along with charts and statistics. You can also export the complete list of errors in Excel format.

MQM Scorecard

Template creation

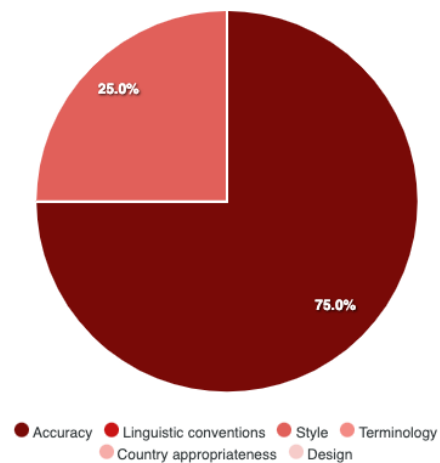
Information	MQM	Feedback
Job Id 595	 Status PASS	General
Translator Id 6	Date Completed 2020-12-16	No Feedback
Reviewer Id 7	Score Details	
Reference Swedish	Achieved Score 97.17%	
Target Swedish	Weighted Err/1K 28.30 ✓	
Word Count 5654	Err/1K 8.84	
	Definitions	
	Max Allowed Weighted Err/1K 300	
	Threshold 1696.20	
	Pass Score 70.00%	

Summary

[Export detailed issue list \(Excel\)](#)

Category	Neutral	Minor	Major	Critical	Error Points
Accuracy	1	2	0	0	70
Mistranslation	1	0	0	0	0
Mishear	0	1	0	0	40
Addition	0	1	0	0	30
Omission	0	0	0	0	0
Untranslated text	0	0	0	0	0
Linguistic conventions	0	0	0	0	0
Grammar	0	0	0	0	0
Spelling	0	0	0	0	0
Typography	0	0	0	0	0
Style	0	0	1	0	90
General style	0	0	0	0	0
Job-specific style	0	0	1	0	90
Terminology	0	0	0	0	0
Wrong terminology	0	0	0	0	0
Inconsistent terminology	0	0	0	0	0

Error Categories - Allocation



Aggregated data on user level

By going to a user's summary page, you can find all their MQM reports. They can be viewed individually, or summarised with the "Aggregated MQM Scorecard" link.

Job Id	Date	Score	Status	Link
595	2020-12-16	97.17%	✓ PASS	View
598	2020-12-16	65.45%	✗ FAIL	View
601	2020-12-16	81.42%	✓ PASS	View
604	2020-12-18	97.40%	✓ PASS	View
609	2020-12-16	81.43%	✓ PASS	View
612	2020-12-18	95.49%	✓ PASS	View
615	2020-12-18	95.84%	✓ PASS	View

Auto create subtitles

This automatic action converts a file from the internal system format into a **deliverable** (e.g. TTML, SRT). The created file will be stored in the project Filezone for easy access by project managers and other users. With additional settings, it can also be made visible to Resources (from a job) and Clients (in Client Filezone).

[If unsure, you might also want to read up on how parenting works with automated jobs](#)

Key concepts

- The source file is the current version of the [online editing file](#) for the job's node.
 - Example: the Auto create subtitles job is linked to part one of a project, and to the German language node.
 - The system will look for an online editing file in the same node
- Any available delivery formats can be selected, including the original JSON file
 - Additional formats are continuously added.
- By combining information about the project as well as language and part information a custom filename can be instructed.
 - E.g. project ID + "_" + language code + "part" + part number

Example use case

- Prepare a subtitling workflow consisting of
 - Translation
 - QC
 - Auto subtitle creation
- When the QC step is done, the subtitles will then be created automatically
- When all subtitles are completed, a project manager
 - enters the project Filezone
 - filters out all deliverables (based on file extension, for instance)
 - batch downloads all files
 - and delivers

Automating delivery

By making a file "Visible to client in filezone" you could allow the client users to fetch completed files themselves. However, that may not be convenient in all cases. By placing an automatic delivery job after the subtitle creation job, subtitles can be automatically sent to an external server (AWS S3, FTP ...) virtually seconds after approval, in the selected format.

Auto FFMPEG render

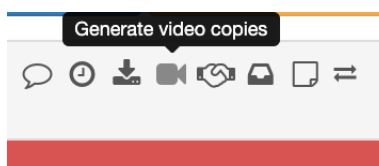
[If unsure, you might also want to read up on how parenting works with automated jobs](#)

This option lets you create deliverables using the files in the project, including the video file. A typical example would be a video with [burnt-in subtitles](#). In order to use this you need.

- A **video** uploaded to the project and set to mode "Standard".
 - It also needs to be linked to the same node/part as the job that the translation is linked to.
 - Typically, this video should be a hi-resolution file *without* burnt-in timecode or text.
 - The simplest way to attach such a file to the project while also being able to work in the online editor is to use the "Media upload" delivery method for a previous job and select a conversion preset in the upload dialogue.
 - This will automatically store the original file with the project *and* create a lo-res proxy for the editor to use, giving them both the correct settings
 - (But adding the necessary files can also be done directly in the Filezone)
- A node with a **subtitle** linked to it
 - I.e. a subtitle file created by online editor
- A **Render preset**
 - This determines the details of the output file.
 - Text styling, video resolution and container, audio format, etc
 - These are edited and created under *Tools & Data* and require some in-depth knowledge to manage
 - The two styling options support:
 - ASS: Italics, Raised, Bold, Underline, Coloured text
 - SRT: Italics, Raised, Bold, Underline
 - In order to use alternative fonts, they first need to be uploaded under *Tools & Data / Fonts*.

By combining these three, Plint can automatically create deliverables as a job.

The same tools and logic can also be applied to render videos "ad hoc" from the page *Generate video copies*, which can be accessed from the project bar.



Use Connection profile(s)

Test bucket (AL) ✕

Deliver files to storage root

No

File matching pattern

.mp3|.docx

All files belonging to the same node as this job that match the pattern will be delivered to the selected location.

Wildcard: +

Multiple patterns: 1

Examples:

- *.txt
- *-episode-*.txt
- the-episode-name*.txt
- the-episode-name*.txt|-episode-*.txt

Auto delivery

Auto delivery is an automated delivery method that can be used in any workflow. When the parent job is done, the Auto delivery will be initiated and files delivered to a predefined location determined by the [Connection profile](#). When you select the "Auto delivery" method, you can select the Connection profile and additional settings on this tab.

Only files linked to the same node (e.g. part/language combination) as the delivery job will be sent to the location. If you have multiple files in that node, you can create a *pattern* to select what files to deliver. The example below shows how two patterns, separated by a pipe character, can be used to target both MP3 and docX files.

What you need:

- At least one *Connection profile* (under Tools & Data).
- A job on a project, with a *parent* job
 - Set this job to Delivery method "Auto delivery"
 - Specify delivery settings on the now visible "Connection profile" tab

[If unsure, you might also want to read up on how parenting works with automated jobs](#)

[General](#) [Editor settings](#) [Casting](#) [Job Fee](#) [Job history](#) [Messages](#) [Notes to self](#) [File settings](#) [Connection profile](#)

Use Connection profile(s)

Test bucket (AL) ✕

Deliver files to storage root

No ▾

File matching pattern

.mp3|.docx

All files belonging to the same node as this job that match the pattern will be delivered to the selected location.

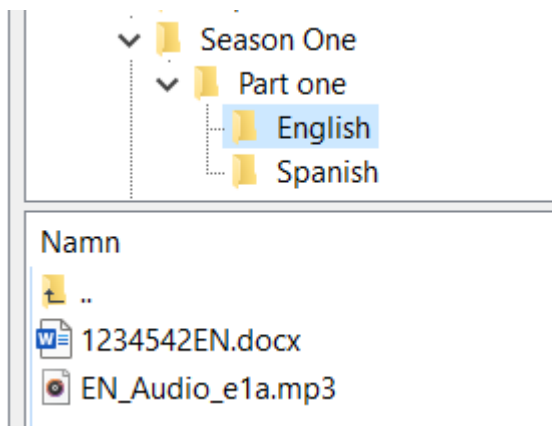
Wildcard: +

Multiple patterns: 1

Examples:

- *.txt
- *-episode-*.txt
- the-episode-name*.txt
- the-episode-name*.txt|-episode-*.txt

Folder structure



If "Deliver files to storage root" is set to "Yes", the files will be sent directly to the location and (optionally) sub folder specified in the Connection profile. If this is set to "No", a folder structure matching data from the project will be created in that location, according to the following pattern: *project name/part label/language name*. (If the parts are unlabelled, all files for the same language will be sent to a single folder).

Auto upload

Auto upload is an automated Delivery method that can be used at any point in a workflow. Each job will monitor a [predefined location](#) (see [Connection profiles](#) for files and ingest them into the system when found, for example a source media file. The file(s) will be linked to the same [node](#) as the job itself and be found in the project Filezone, just like any upload. You can use dynamic filename matching patterns including wildcards to fetch the right file, for example media for a specific part or episode. By adding language code as a criteria you can be even more specific.

The fetched files can be given the same properties as when editing them in the Filezone, such as "Visible on Files tab". They can be set to [Media files](#) to be used in the online editor, or Standard files if they are to be delivered or processed further.

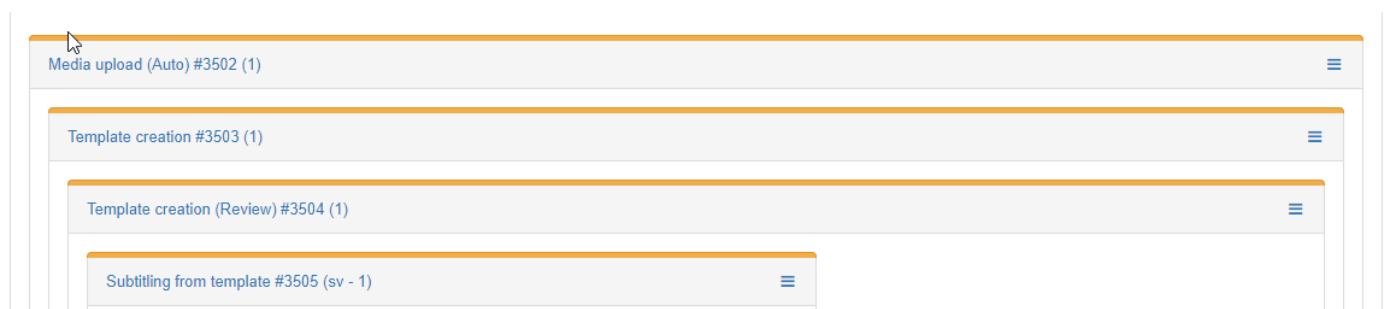
What you need:

- At least one *Connection profile* (under Tools & Data).
- A job on a project
 - Set this job to Delivery method "Auto upload media"
 - Select Connection profile and other settings on the now visible "Auto upload" tab

How it works: At regular intervals (every 10 minutes) each connection profile with attached Auto upload jobs will be checked for matching files. (Please note that the option to specify the checking frequency has been deprecated as of fall 2021.)

Auto upload jobs need to have status "In process". They can be paused or aborted by changing job status. When files are found and fetched from the selected location, the job in question will change status to Done, which will stop the task and open any child job(s) according to normal Plint logic.

*A workflow starting with an automated **Media upload** job. When a source media file is available in the selected location, it will be automatically ingested, launching the rest of the workflow.*



The settings tab for an Auto upload job. In this example, files whose name include the part media id matching this job will be found and ingested as Media.

Connection

Connection profile (required)

Test S3 Bucket x

Rules

File matching pattern

source_(part_media_id)*.mp4

Any files that match the pattern will be used.

Wildcard: +

Multiples patterns: !

Examples:

- *.mp4
- *-episode-*.mp4
- the-episode-name*.mp4
- the-episode-name*.mp4|*-episode-*.mp4

You can specify a dynamic filename pattern using the following variables:

(part_label) = Episode name
(part_number) = 2
(part_media_id) = ABC123
(language_code) = EN-gb
(project_title) = The TV Series
(project_id) = API123
(project_api_id) = PID_API123

File settings

File mode

Media

File category

Video

Visible on Files Tab

No

Visible to clients in filezone

No

Visible to child projects

No

Convert after upload?

Upload without conversion

Timecode start time

HH:MM:SS:FF

ASR

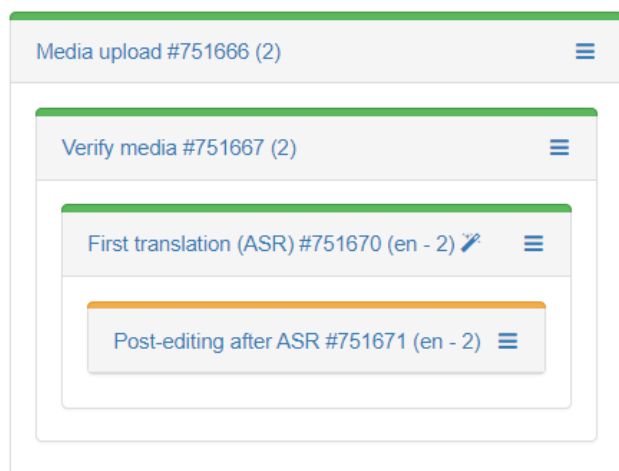
ASR availability will depend on site settings

[If unsure, you might also want to read up on how parenting works with automated jobs](#)

The "ASR" Delivery method sends the **audio** of the current media for the node to an external service for **Automatic speech recognition**. The result will be parsed into Plint's internal subtitle format and attached to the current node. This means that a subsequent job in [Plint Subtiter](#) can automatically load this file for further work, such as post-editing. The file could also be automatically converted to a delivery format and used right away, although this is rarely recommended with raw ASR output.

A typical setup:

- A project with the **correct source language** specified
- Attach/upload **media** to each part, in typical fashion
- After the media steps (i.e. upload/convert/verify), **add an ASR job**
 - This job can be attached to the part itself, or to a target language
- Follow this with a **online editor job**, for reviewing and adjusting the file



In this example, once the media has been verified and the user changes status, the ASR job will trigger automatically. When this process is completed, the ASR job will automatically change status and the review/post-editing job will become open for work. A subtitle milestone will be stored and can be seen under *Review / Track changes* to enable easy comparison between the original and final versions.

Settings

There are three main settings which can be applied to the ASR job:

- **Service**

- Depending on what is enabled for your platform, options may include "Azure", "AWS", and others.

- **Segmentation**

- **Rule-based** segmentation will take into account the [Report tool](#) rules matching the job when splitting the text into individual subtitles, basing the subtitle length on the minimum/maximum duration setting.
 - The *Basic* variant will only take the *duration* setting into account when deciding where to end a subtitle. This may require increased post-editing.
 - The *Improved* variant takes longer to process but will reduce the risk of single words ending up at the beginning or end of subtitles.
- Whereas **Sentence** segmentation will allow longer clips and base the clip length on **punctuation**.
 - This is suitable for non-subtitle work and for when you prefer to do the splitting at the post-editing stage, for example using Plint Subtitler shortcuts.
- With both types, the *Clip separation* rule matching the project/job will always be applied to ensure that correct spacing is used between subtitles.

- **Speaker recognition**

- This option will attempt to use the third party service's speaker recognition functionality. The result can be stored either as the first line of the subtitle, or in the Annotation field. The latter field can be either visible, hidden or editable in Plint Subtitler, based on job settings.
- Please note that with Rule-based segmentation, it is common to get more than one speaker label with the same subtitle. They will all be saved on a single line.
 - Example (Rule-based segmentation, Speaker label stored as part of Subtitle)

Index	In-time	Duration	Out-time		Count	Reading Speed	Gap
01	00:00:03:26	01:16	00:00:05:12	⚠	10 22	20.8	+ 00:03
Speaker: 2							
Millions of years ago,							

ASR settings tabs

Segmentation (required)

Rule-based (Improved, slower) ▾

Service (required)

Azure ▾

Speaker recognition (required)

Off ▾

Things to note

- Depending on the segmentation type, certain values will be mandatory in the Report preset in order for the job to run properly. To ensure complete functionality with rule-based segmentation, make sure the Report preset has settings for:
 - Subtitle separation (formerly *Clip separation*)
 - Duration (min and max)
 - Maximum characters per line
- The ASR process can also be run "ad hoc" using the microphone icon in the joblist
- An ASR job set to "Awaiting corrections" will *not* run automatically
 - This status is used to indicate that something went wrong with the process and something will probably need to change before another attempt is made.
- Ensure that the service you select supports the current source language, or the process will fail.
 - For AWS (*Amazon Transcribe*), the list can be [found here](#).
 - For Azure (*Speech to text*), [see here](#).

Automated jobs and job parenting

Understanding the following concept is essential when using automated jobs:

For all automated jobs except "Auto upload", the action is triggered when the **parent job changes status**. Status change needs to be done *through* the parent job, for example by clicking "Report as done" on a Report job or through the "Change status" dialogue in the online editor.

Simply editing or batch editing the properties of a job and setting status to "Done" will *not* trigger automatic jobs.

Typical examples:

- Media verification in Plint Subtitler, followed by "ASR"
- Final QC in Plint Subtitler, followed by "Auto subtitle creation"
 - and perhaps "Auto delivery"

KNP / Term bases

In order to ensure consistent terminology, Excel sheets with termbases (aka Key names and phrases) can be uploaded into Plint Core, linked to projects and used within Plint Subtitler as a live help while translating.

Adding KNP to a project includes these steps

- Create or update an Excel file with terms and phrases in all relevant languages
- Import this file into Plint Core
- Link it to a "Production KNP"
 - This could be anything from a certain client, a topic or a TV show – anything with a specific terminology.
- Create a project linked to the same Production KNP
- Include a step in the workflow to add KNPs to the JSON work file

The Excel file

The file needs to follow a certain format (a template is available on request). The second and third columns (source and target language) are mandatory. The rest can be used for additional information, for example when working on a shared Excel book.

	1	2	3	4	5
1		Production Title			
2	TYPE	SOURCE	SWEDISH	EPISODES (S00E00)	COMMENTS
3	TERM	Batman	Läderlappen		The old Swedish name for Batman
4	TERM	Catwoman	Kattdamen		The translation used in this series
5	PHRASE	Holy breaking and entering	Rena rama inbrottet		Use this translation for this Robinesque phrase

Specifying language

Each target language requires a corresponding tab in the file. The most important part is to specify the language code in quotes on this tab, e.g. "sv". This code must match the target language on the project exactly, including locale if used. ("sv-SE" will for example not match "sv").



Importing the file into Plint Core

Under "Tools & data", go to "Production KNP". This is always available to Workbench users and can be added for Manager roles. To introduce a new "Production KNP", click Create. You can also edit an existing one.

Please note that uploading a new file to an existing KNP will remove all current terms and phrases before adding new ones. This is to ensure that outdated terms are not used.

Dashboard / Tools & data / Production KNP

Available Production KNPs

Create a Production KNP +

Name	Updated	Languages	Description	
Template	2023-12-19	4		

Total number of available Production KNPs: 1

Upload the Excel file and check the result before saving.

Creating Production KNP

SaveCancel

Dashboard / Tools & data / Production KNP / Create KNP

Name (required)

Acme KNP

Description

Production KNP file

Choose fileKNP example.xlsx

Select the KNP-file you wish to upload.

Successful file upload! Do not forget to save your changes.

Production KNP Information

Languages included4

Danishda

Finnishfi

Norwegianno

Swedishsv

Link the KNP to the project

Project details & descriptionInputOutputPrice & internal informationTarget specification

Project title (required)

The KNP project

Status

New order

Client company

— N/A —

Project type

— N/A —

Project Forum?

Yes

Tag

Project description

Project manager

Workbench user

Client contact

— N/A —

Parent Project

— N/A —

Production KNP

— N/A —

— N/A —

Template

Acme KNP

Cli

De

Co

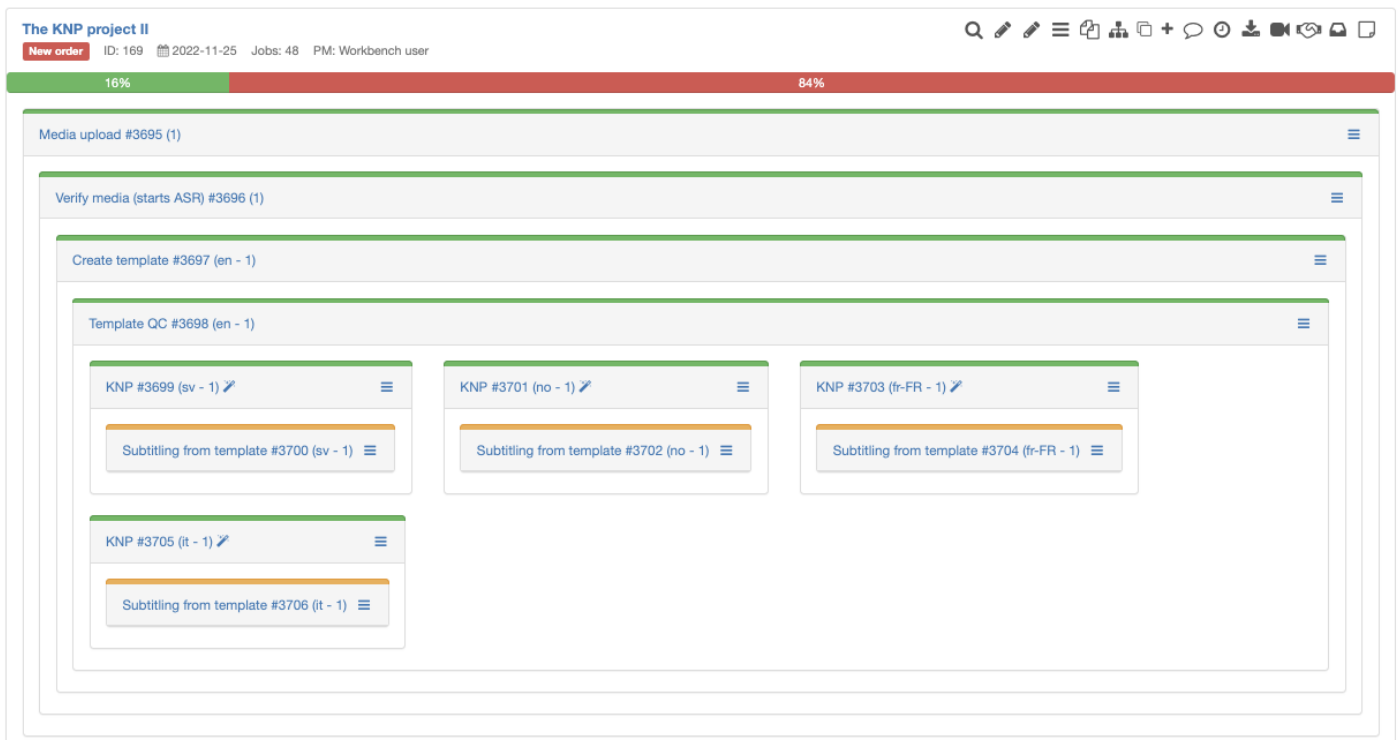
Pr

Dis

Add KNP jobs to the project

KNP jobs are automatic jobs that add information to the work file. They must be run when the template text is in place but before the subtitler starts working in Plint Subtitler. This is typically solved by adding these jobs to a joblist template, but KNP jobs can be added manually for testing, as long as the workflow is intact.

In this example, the KNP jobs have already run, triggered by the status change when the English template was completed. The jobs need the Delivery method "KNP" and what they do (besides importing the template if needed) is to augment the JSON file with suggestions based on the reference text.



Seeing and using the result in Plint Subtitler

We now have a work file prepopulated with the reference text and suggestions for how certain terms are to be translated.

When the user moves to a block with such a term, it is highlighted. In order to simply insert the suggested term, just click on it in the list under "Production KNP".

<-

1

>

Auto save is disabled, due to you have opened someone else's job.

00:00:07:06

00:00:00:00

00:00:39:12

01

00:00:07:06

00:00:10:13

+03:07

A ticking bomb means trouble for Batman and Robin!

02

00:00:11:18

00:00:13:17

+01:23

Holy breaking and entering, it's Batgirl!

02

00:00:13:19

00:00:16:02

+02:07

Quick, Batgirl. Untie us before it's too late.

08

00:00:16:04

00:00:18:13

+02:09

It's already too late.

03

00:00:18:15

00:00:21:13

+02:22

I've worked for you a long time and I'm paid less than Robin.

09

00:00:21:15

00:00:25:10

+03:19

Same job, same employer means equal pay for men and women.

04

00:00:25:12

00:00:29:16

+04:04

- No time for jokes, Batgirl.
-It's no joke. It's the federal equal pay law.

05

00:00:29:18

00:00:33:13

+03:19

- Holy Act of Congress!
- If you're not getting equal pay

Index

In-time

Duration

Out-time

Count

Reading Speed

Gap

#

Production KNP

01

00:00:07:06

03:07

00:00:10:13

0

0

+ 01:05

1

Läderlappen

A ticking bomb means trouble for Batman and Robin!