QADistiller Integration

AUTHOR

Sufian Reiter
Sufian.Reiter@plunet.com

DATE

2023-06-12

Version History

Date	Version	Author	Changes
23.09.2020	1.0	Sufian Reiter	

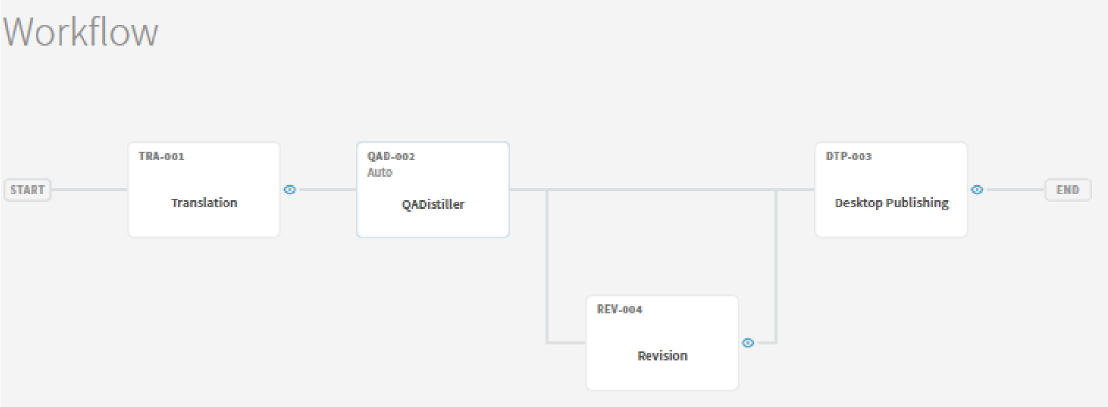
Table of Contents

1. Short Description

Yamagata's QA Distiller (http://www.qa-distiller.com/en) is a free stand-alone tool for finding translation errors in bilingual files. The solution has a so-called batch mode function: the QA check can be carried out automatically from the command line without a user interface, provided that all the necessary parameters are also transferred.

The Plunet BusinessManager has an ApplicationManager module with which additional programs can be called up from the command line. This is done via an automatic job in a workflow.

Consequently, it is possible to set up the QA Distiller as an automatic job in Plunet and, depending on the result of the QA check, to let the workflow continue, stop it or trigger different follow-up jobs.



In the sample workflow above, the revision job is only started when a certain number of errors is reached. Otherwise, the workflow continues directly with desktop publishing.

2. Requirements

2.1. Technical Requirements

Plunet BusinessManager 7.4+ and additional module ApplicationManager

Yamagata QA-Distiller 9* (https://www.yamagata-europe.com/en-gb/qa-distiller)

*last successfully checked version

3. Configuration

3.1. Yamagata QA-Distiller

3.1.1. Installation

The QA Distiller must be installed on the server. The storage location should normally be `%ProgramFiles(x86)%\Yamagata\QADistiller9\`.

3.1.2. Settings / Profiles

The settings or profiles for later processing should be defined for each customer. For this purpose, the QA Distiller normally uses a Userlists folder. You can find more information here:

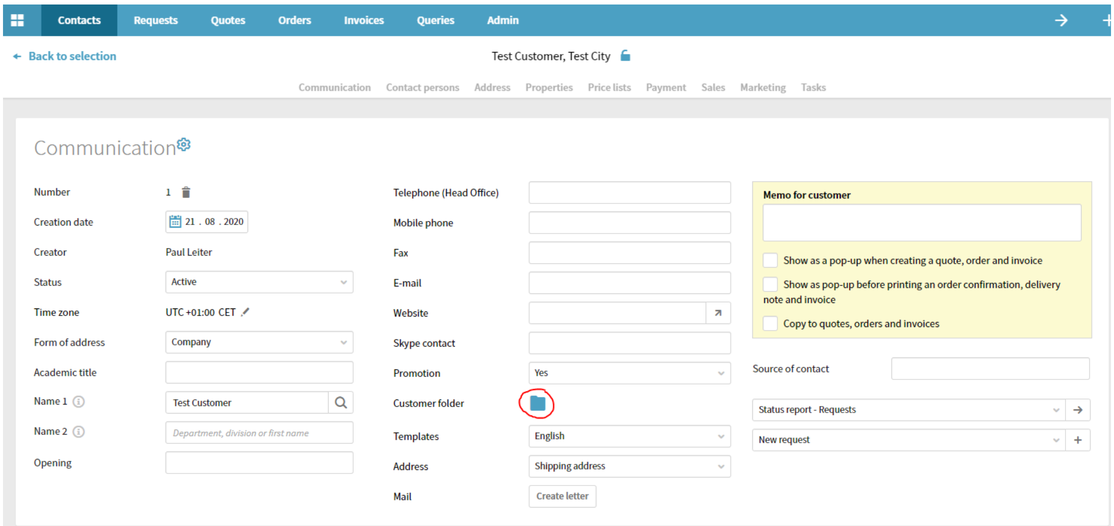
https://www.qa-distiller.com/help/en-us/concept/co_about_user_lists.html

Attention: At present, only one profile per customer can actually be used by the integration

3.2. Plunet BusinessManager

3.2.1. Customer Profile

Each customer has his own folder in Plunet BusinessManager.



Test Customer, Test City

Communication

Number	1	Telephone (Head Office)	
Creation date	21 . 08 . 2020	Mobile phone	
Creator	Paul Leiter	Fax	
Status	Active	E-mail	
Time zone	UTC +01:00 CET	Website	
Form of address	Company	Skype contact	
Academic title		Promotion	Yes
Name 1	Test Customer	Customer folder	
Name 2	Department, division or first name	Templates	English
Opening		Address	Shipping address
		Mail	Create letter

Memo for customer

□ Show as a pop-up when creating a quote, order and invoice

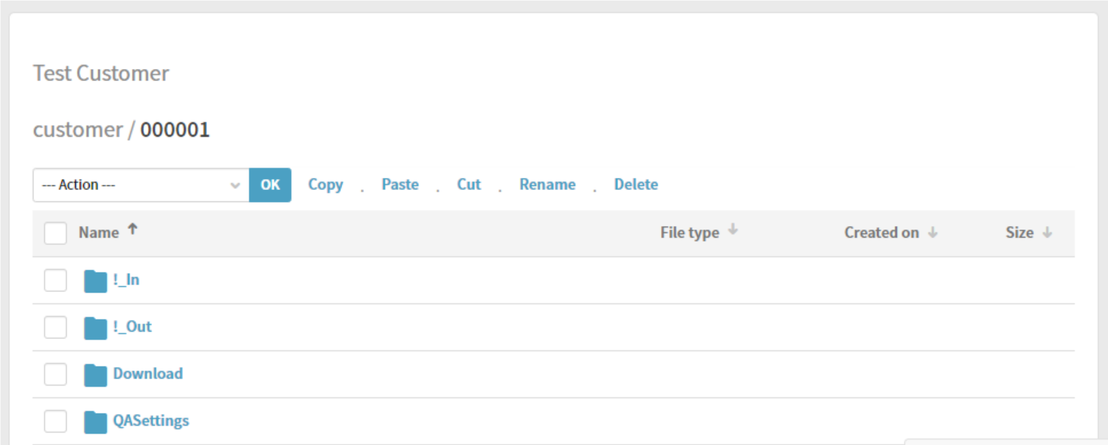
□ Show as pop-up before printing an order confirmation, delivery note and invoice

□ Copy to quotes, orders and invoices

Source of contact

Status report - Requests

New request



Test Customer

customer / **000001**

□	Name	File type	Created on	Size
□	!_In			
□	!_Out			
□	Download			
□	QASettings			

The customer specific profile (QADistiller Userlist folder) should be stored here. The folder must have the same name in every customer profile (e.g. QASettings).

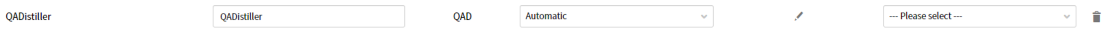
3.2.2. Storing the script in the Plunet directory

The script to start the process must be stored in the Plunet document directory (where the folders Order, Customer, etc. are located). Ideally you name this folder App and put the qadistiller.bat file there.

If you decide to use a different path or file name, this has to be adjusted in the definition of the program call under 3.2.3.

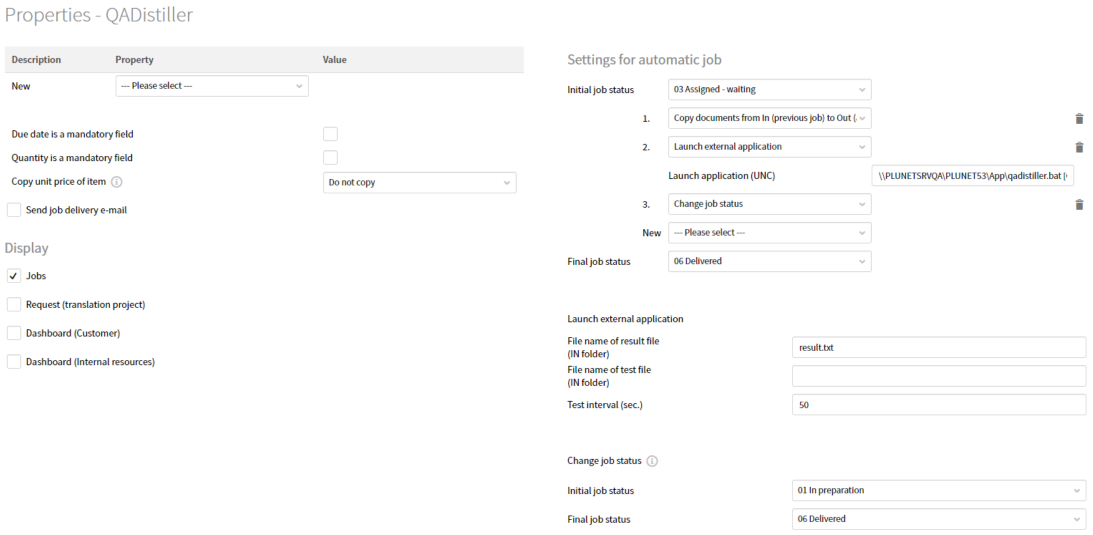
3.2.3. Automatischen Job für QADistiller anlegen

Go to the Admin|Jobs area and create a new job for the QADistiller (job category: automatic)



QADistiller	QADistiller	QAD	Automatic	--- Please select ---

Using the pencil icon you can now open the settings below.



Properties - QADistiller

Description	Property	Value
New	--- Please select ---	
Due date is a mandatory field		□
Quantity is a mandatory field		□
Copy unit price of item ⓘ		Do not copy

□ Send job delivery e-mail

Display

☑ Jobs

□ Request (translation project)

□ Dashboard (Customer)

□ Dashboard (Internal resources)

Settings for automatic job

Initial job status: 03 Assigned - waiting

1. Copy documents from In (previous job) to Out (
2. Launch external application

   Launch application (UNC): \\PLUNETSRVQA\PLUNET53\App\qadistiller.bat [
3. Change job status

New: --- Please select ---

Final job status: 06 Delivered

Launch external application

File name of result file (IN folder): result.txt

File name of test file (IN folder):

Test interval (sec.): 50

Change job status ⓘ

Initial job status: 01 In preparation

Final job status: 06 Delivered

The following settings must be made for the Autojob in any case, as they are expected by the QADistiller.

- Initial job status: 03 Assigned and Waiting
- Copy documents from IN (previous job) to OUT (automatic job)
- Launch external application

Launch application(UNC):
pathwerethefileis|nameofthebatchfile.bat [CustomerID] [JobNo]
(Example: *\\PLUNETSRVQA\PLUNET15\App\quadistiller.bat [CustomerID] [JobNo]*)

- Change job status
- Final job status: 06 Delivered (or 07 Approved)

The following setting should also be made for the external program:

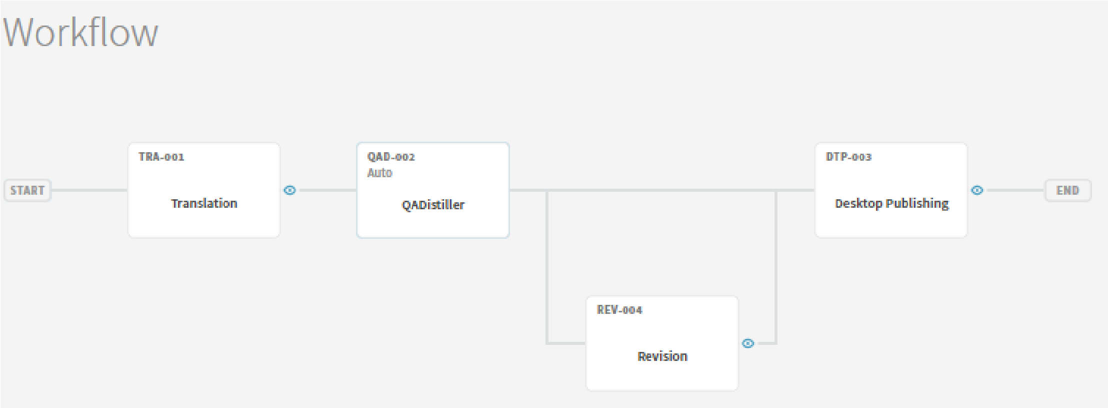
- File name of the result file (in IN directory): result.txt

3.2.4. Workflow Setup

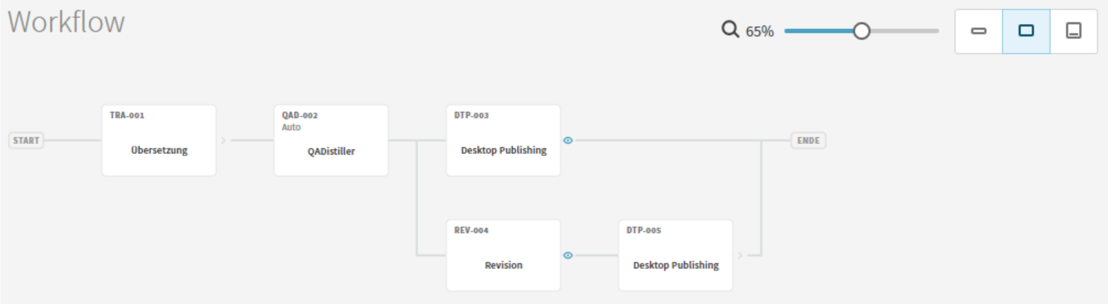
Under Order|Workflow you can define a new workflow.

The following examples are possible

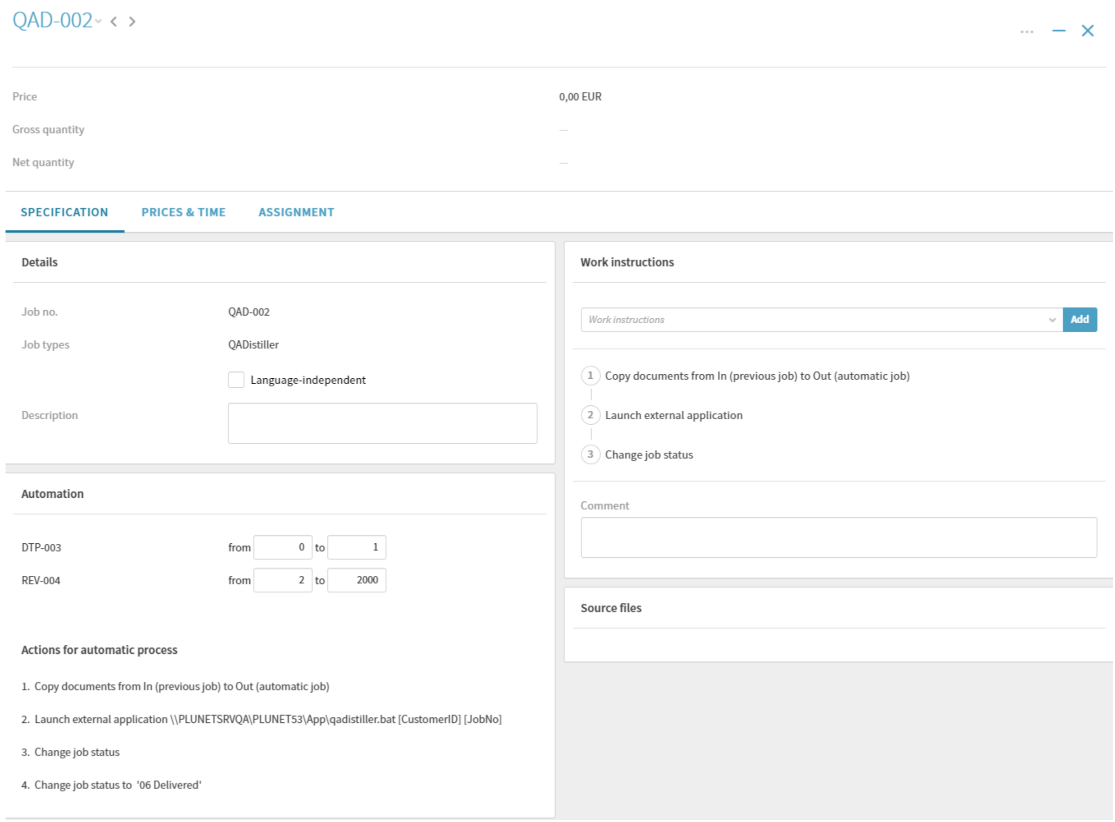
Example A:



Example B:



Since the QA check delivers a result with the number of errors, you can now work with a conditional workflow. To do this, open the QADistiller Job.



QAD-002

Price	0,00 EUR
Gross quantity	–
Net quantity	–

Details

Job no.	QAD-002
Job types	QADistiller
	□ Language-independent
Description	

Automation

DTP-003	from	0	to	1
REV-004	from	2	to	2000

Actions for automatic process

1. Copy documents from In (previous job) to Out (automatic job)
2. Launch external application \\PLUNETSRVQA\PLUNET53\App\qadistiller.bat [CustomerID] [JobNo]
3. Change job status
4. Change job status to '06 Delivered'

Work instructions

1. Copy documents from In (previous job) to Out (automatic job)
2. Launch external application
3. Change job status

Comment

Source files

In the Automation section you can define the error ranges. The example defines that up to 1 error, the workflow should continue with desktop publishing. From 2 errors on, the revision step should be performed first.

Please make sure that the subsequent jobs are set to start with the appropriate start signal (e.g. previous job status change to *delivered*).

The configuration is now complete.

4. Working with the workflow

4.1. Source Files

To start the QA Distiller job, a bilingual .ttx or .xliff file must have been delivered in the previous job (IN folder). Alternatively, a bilingual file can also be manually stored in the OUT folder of the automatic job, which would then be taken into account *(more suitable for testing purposes)*.

4.2. Project specific terminology data

If the predecessor job (IN folder) or the OUT folder of the automatic job contains a terminology file in .dict format that is specific to the respective project, the QA Distiller automatically takes this into account.

4.3. Application workflow

The QADistiller Job will be started automatically, if the previous step is set to delivered (depending on the settings this may vary).

The following work steps are then performed:

- The delivered files of the previous job are copied into the OUT folder of the QADistiller job
- The script is started and the parameters CustomerID and Job Number are passed along
- Tasks of the script:
- A configuration file relevant for the QADistiller is created for the project (nomenclature: job number.qadbatch) with the following information:
- Folder where the customer profile for QADistiller is stored
- Bilingual files to be checked including storage location
- Terminology data to be considered in .dict format including storage location
- Output formats to be created and their content
- Display of progress (should always be set to 0)
- QADistiller is started by transferring the configuration file
- Script checks when the QADistiller is finished and has stored all generated files in the IN folder of the QADistiller job
- Unneeded files are moved to a subfolder QAResults (so that they are not available in for the resource of the follow-up job
- The CSV file of the QA result is analyzed, the total number of errors is written into a result.txt, which can later be used for the conditional workflow

- The original bilingual data and terminology data as well as the QA result as HTML file are moved into the IN folder of the QADistiller job (they are automatically available for the resource of the follow-up job)

- The QA Distiller job is automatically set to *delivered*, the "relevant" follow-on job is automatically started, if *assigned and waiting*.