

## Bypassing Content Creation

Many OL Connect jobs require making simple changes to print-ready batches of documents: adding OMR marks, commingling jobs, postal sorting, and so on. For those types of jobs, no merging of any data is required and yet, after data mapping is done, you must still go through the Content Creation step before moving on to Job Creation. That's a waste of time and CPU clock cycles. In this article, we'll explore a new REST call available in OL Connect 2020.1 that allows a job to completely bypass Content Creation, and we'll see how overall performance is thereby improved.

### The use case

Say you have a simple PDF that contains a series of variable length documents. And all you want OL Connect to do is add OMR marks to each document inside the PDF. The standard way of doing this from OL Connect Workflow would be to use either the 4 separate tasks (*Data mapping, Content Creation, Job Creation, Output Creation*) or the *All in One* task.

If performance is key, or if you are not using Workflow, then instead of using the native Workflow tasks you might script the process by using the OL Connect REST API. Again, you may opt to script all separate tasks or just the All in one task. Either way, you will be performing Content Creation even though no data is actually being merged onto the original PDF (remember that OMR marks are added at the Output Creation stage, not during Content Creation).

With OL Connect 2020.1, the REST API offers a new procedure that performs the Data Mapping as usual and also immediately creates a Content Set whose content is the original PDF file itself (instead of having the Merge Engine create a new PDF). That allows the Job Creation operation to be called immediately after this new Data Mapping operation is complete. Knowing that even the simplest of Content Creation processes is almost always the bottleneck in creating any OL Connect job, bypassing that step results in significant improvements to overall performance.

### The tests

In preparing this article, we started from the Generic Invoice demo available from our Resource Center (<https://help.objectiflune.com/en/#pps-sample>). Instead of using the demo's template, we created a very basic template that does nothing more than map the input PDF to the background of the template. The Job Preset sorts the various documents and the Output Preset stamps OMR marks on each document.

We then created a Workflow process that takes an incoming PDF and runs all tests sequentially. The words in (parenthesis) are the code names used in the results for each test:

1. Using the 4 default Workflow tasks (**TASKS**)
2. Performing all 4 steps again, through a single script that uses the REST API (**REST**)
3. Performing the All in One procedure, through a single script that uses the REST API (**REST AIO**)
4. Performing the new Data Mapping procedure, followed by Job Creation and Output Creation, through a single script that uses the REST API (**REST W/O CC**)

We used 3 input data files to perform the tests: a small 16-page PDF, a larger 160 page PDF and then an 800-page PDF. For each input file, we ran the tests 3 times in a row and the results presented here are an average of all test runs for each input file.

All resources are available in the attached ZIP file (see the **Resources** section at the end of this article).

### The results

Because the overall speed of each test may vary greatly depending on the underlying hardware, we normalized the results over an arbitrary value of 100 *units of time*: the slowest procedure is deemed to have taken 100 units of time, and all other procedures are therefore a percentage value of that slowest procedure. Once again, those units *are not* milliseconds, seconds or minutes. This should allow you to compare your own results if you run the same tests on your machine or on a VM.



Even with a small job, the difference in performance is significant. The new REST call essentially cuts down in half the time it takes to process the job using the standard OL Connect Workflow tasks.



With a larger job, the gap widens between the first two columns which perform all 4 separate steps (either as tasks or as separate REST calls) on one side, and the last two columns which perform optimized calls (All In One and the new DM call).



With much larger jobs, the gap widens to the point where the script using the new REST call takes less than a quarter of the time required by standard OL Connect Workflow tasks. To put it in perspective: a one hour process using standard tasks would take around 15 minutes to complete!

#### Additional notes

- You may have noticed that the first two columns remain stable relative to each other regardless of job size: that's because they perform (almost) the same set of actions. Why, then, are the OL Connect Workflow tasks slower than their REST equivalent? Well, that's mostly due to metadata, which is created and updated by the tasks whereas the script doesn't generate any. There is also a slight bit of overhead from having 4 separate tasks instead of a single script, but that's not statistically significant.
- The larger the job is, the more the last two columns distance themselves from the first two. That is also expected because with larger jobs, the streaming process used by the All in One procedure (from DM to CC) has time to make use of its multiple threads. It also explains why the performance of those last two columns relative to each other remain fairly consistent regardless of job size. But still, the new DM call that bypasses Content Creation is always significantly faster.
- The scripts in the Workflow process all use the Enhanced JScript language, which explains for instance why the *btoa()* method used in the authentication REST call doesn't have to be polyfilled in each script. And that makes for much more readable code.
- Documentation and syntax information for the new REST call will be available in the REST API reference when OL Connect 2020.1 is released. If you already want a glimpse at how it works, open the 19th task in the process and scroll down to line 65 of the script.

**The resources**

This zip file contains all the resources required to run the tests. The folder structure used is the same as that of the Generic Invoice sample, so you may want to install that before hand and then extract the zip file in the proper location. You will also have to import in OL Connect Workflow the basic template that's required for the Content Creation task and API calls.