Output Agent Throughput

Best Practices

Version: 7.3.x

Written by: Product Knowledge, R&D
Date: September 2020

Hyland

© 2015-2019 Hyland Software, Inc. and its affiliates.

Table of Contents

Output Agent Throughput **4**
About Output Agent **4**
Environment and configuration performance 4
Remote installation *4*
Keyfile polling interval *4*
Export directory cleanup *4*
File Type and Annotation Performance 5
Source file type *5*
Destination file type *6*
Annotations *6*
Document Size and Multi-Page Exporting 7
Multi-page destination formats *7*
Scaling Output Agent 8
Multiple agents on a single node *8*
Agents on additional nodes *8*

Output Agent Throughput

For customers who process high numbers of pages, Perceptive Software recommends tuning systems for high throughput and efficiency to maximize performance in Output Agent. This document provides information to help you determine your Perceptive Content throughput performance needs based on the volume of documents processed by your organization.

About Output Agent

The dedicated and high-volume, server-side Output Agent exports and prints documents from Perceptive Content. It supports a wide variety of source file types, including TIFF, PDF, and DOC. When exporting image file types, Output Agent can process annotations as well as change the DPI and color depth of files. Output Agent controls the destination properties of an export or print job using output profiles.

Output Agent only accepts jobs from either a keyfile or job submission. Keyfiles are plain text files that contain a search filter that Output Agent uses to determine the documents to export. The default Output Agent profile located in the inserverOutput.ini configuration file processes documents matching the search filter. Users can submit jobs with an included iScript that has instructions to change the output profile on a per-submission basis. Refer to the Perceptive Software Customer Portal for complete documentation on Output Agent configuration.

Environment and configuration performance

The deployment and configuration settings impact the performance of Output Agent. The following behaviors are crucial to achieving high Output Agent throughput.

Remote installation

Output Agent always operates in remote mode whether installed locally to the Perceptive Content Server or on a separate host. Your system does not benefit from installing Output Agent locally unless the network quality is not optimal between the Perceptive Content Server and a remote host. Document export and storage are resource intensive and can cause Output Agent to compete with other Perceptive Content services if installed locally. We recommend installing multiple instances only on a dedicated remote host. It is important to note that printing and DICOM features of Output Agent are only available on Windows installations.

Keyfile polling interval

Output Agent monitors a directory for new keyfiles at a set interval of seconds, controlled by the inserverOutput.ini setting poll.interval. The agent waits for the interval after each successful output job, including jobs submitted by iScript. If output jobs are primarily created by iScript, we recommend that you set poll.interval=1 to reduce time waiting for keyfiles.

Export directory cleanup

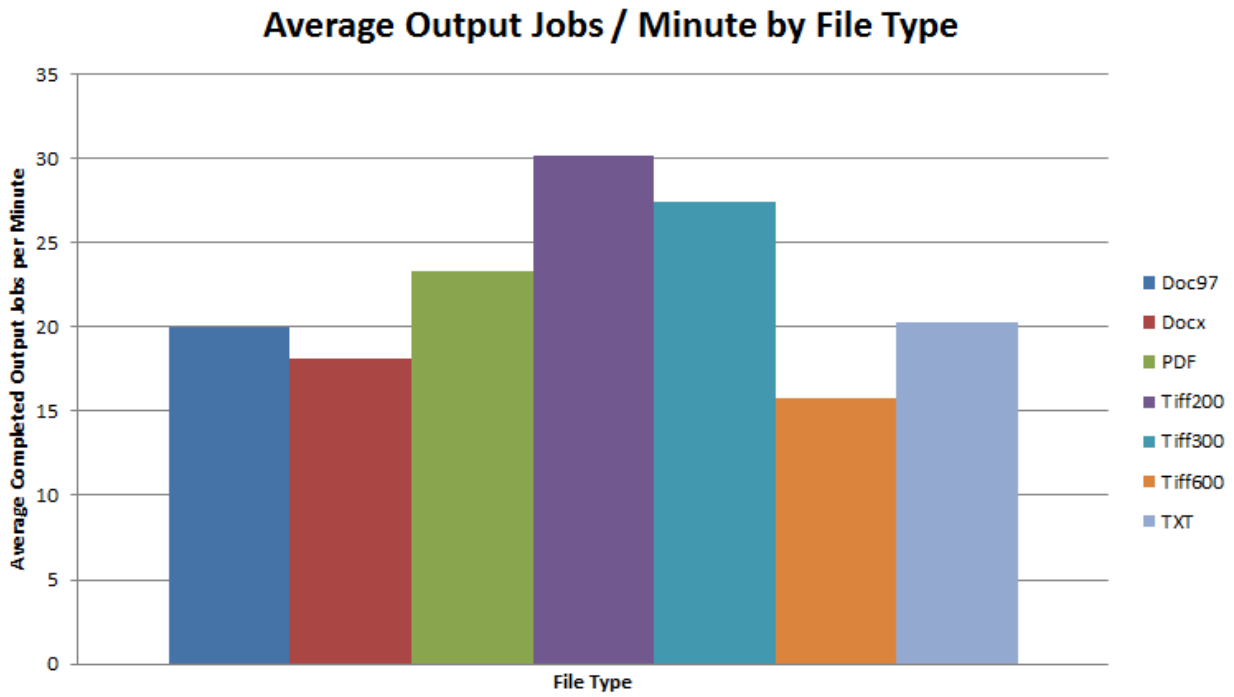
Output Agent sends all exported documents to a configurable directory. For business processes involving high export job volume, this directory's contents can quickly expand, becoming inconvenient for the user and operating system. A large number of export files can negatively impact performance. We recommend periodically moving exported documents to a directory not used by Output Agent.

File Type and Annotation Performance

Output Agent supports a variety of source and destination file types. These files types affect the performance and throughput of Output Agent as well as the system resources Output Agent requires. For our testing standards for these performance tests, we used 40,000 identical files, differing only by file type.

Source file type

Output Agent performance varies with source file type. Testing included the most common file types seen in customers' environments. Three different dpi levels were used for TIFF files (200, 300, and 600) to simulate basic scanned images, OCR-ready images for Recognition Agent, and high resolution. The text based formats tested are PDF, TXT, DOCX, and DOC files.



The test results show that a 200 and 300 dpi TIFF files have the highest throughput. Output Agent still re-renders the image even when the source and destination file types are TIFF. Large TIFF files, such as 600 dpi files, have an exporting performance price that is approximately half the throughput of exporting 200 dpi TIFF files. The resource usage for each of the file types roughly corresponds with their throughput, with the best-performing file types using the fewest resources.

Extension	CPU%	Memory (MB)
TXT	8.75	27
PDF	9.75	28
Doc97	10.75	28
DOCX	10.25	30
Tiff200	7.25	18
Tiff300	9.5	31
Tiff600	13.5	103

Destination file type

Users can optimize Output Agent performance when exporting to PDF and TIFF file formats.

- When exporting to TIFF format, select the TIFF LZW or TIFF G4 format.
- When exporting to PDF format, select the PDF raster format with G4 compression.

These formats yield a 50% throughput improvement when compared to other available compressions. For a complete list of supported export formats, refer to the *Output Agent Installation Guide* or reference Product Documentation on the Perceptive Software Customer Portal.

Annotations

Annotations are important forms of metadata for a document or page. Output Agent supports the output of annotations from any annotation-supported file type. Exporting annotations incurs a performance penalty of approximately 30% on 300 dpi TIFFs. There is no performance impact corresponding to the number of annotations; the penalty depends solely on whether annotation exporting is enabled.

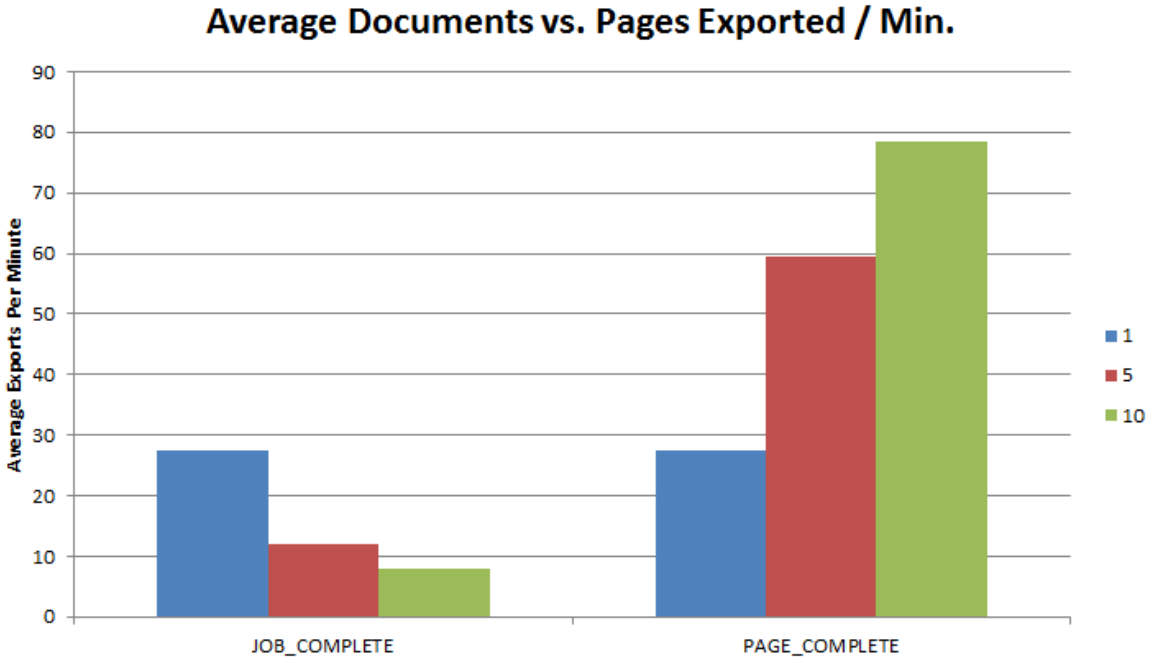
Document Size and Multi-Page Exporting

Perceptive Content supports documents with multiple pages, and these pages can be files with various formats. This flexibility means that, while export jobs with Output Agent are submitted at the document level, the agent must process each page of the document individually to correctly convert from one file type to another, causing some slight differences in the rate of export jobs completed.

Multi-page destination formats

Multi-page exporting enables Output Agent to export all pages of a document into a single multi-page file. Each additionally exported page of a document increases the time needed for Output Agent to complete the export job. These additional pages decrease the document throughput of Output Agent but increase the page throughput. Exporting one five-page document requires less overhead than exporting five one-page documents.

In this example, the document throughput is decreased by 60% and page throughput is increased by 100%. Results are shown for documents containing 1, 5, and 10 pages.



Scaling Output Agent

Users can implement additional instances of Output Agent to increase overall throughput. Multiple instances of Output Agent are supported for single hosts and for across multiple hosts. The number of agents per host is only limited by system resource availability. Each instance of Output Agent requires a separate license.

Multiple agents on a single node

Instances of Output Agent scale linearly. Where N is the throughput of one Output Agent, each additional instance of Output Agent on a node yields an additional 1N throughput. The number of Output Agent instances per node is only limited by hardware and network resources. The CPU and memory consumption of each additional agent also scales linearly. For example, if a single Output Agent is consuming an average of 10% CPU, adding three additional instances consumes an average of 40% CPU.

Agents on additional nodes

Like scaling Output Agent instances on a single node, multi-node instances of Output Agent scale linearly. Scaling Output Agent by node has additional benefits over a single node solution, including failover redundancy and the ability to scale throughput beyond the hardware resource limitations of a single node.