# EMC® CAPTIVA® CAPTURE

## Version 7.5

## Performance Sizing and Tuning Guide

EMC Corporation
*Corporate Headquarters*
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

**Documentation Feedback**

Your opinion matters. We want to hear from you regarding our product documentation. If you have feedback about how we can make our documentation better or easier to use, please send us your feedback directly at ECD.Documentation.Feedback@emc.com

# Table of Contents

# List of Tables

# Chapter 1 Overview

## Introduction

This document provides system administrators and members of the Information Technology department with sizing and tuning considerations for the Captiva Capture system. It summarizes the benchmark testing performed on different Captiva Capture components and provides guidelines to help size a Captiva Capture system to meet individual performance needs.

Benchmark test results and sizing information are provided for each of the components: the InputAccel Server, the InputAccel Database, a subset of Captiva Capture client modules, Production Auto-Learning Supervisor, and the Captiva Administrator. Discussion areas include information about the test environment and the testing methodology used to extract the benchmark results, summary of benchmark results, sizing and tuning recommendations, and a list of critical factors that impact performance.

The benchmark testing was performed in a controlled test environment. These results may vary for your specific production environment. Use the results and recommendations presented in this document as a starting point to help determine the appropriate sizing and expected performance of the Captiva Capture system for your specific production environment.

## What's New

The following sections in the guide contain the new information and updated benchmarking results:

- InputAccel Server results: refer to section InputAccel Server.
- Classification module results: refer to section Classification.
- Extraction module results: refer to section Extraction.
- Standard Import module results: refer to section Standard Import.
- Captiva Web Client and REST services: refer to section Captiva Web Client and REST Services.
- Completion module over WAN: refer to section Completion over WAN.
- Identification module over WAN: refer to section Identification over WAN.

# Chapter 2 InputAccel Server

## Introduction

The InputAccel Server is the central point through which all data coming from and sent to the client modules must flow. The InputAccel Server coordinates, prioritizes, schedules, and routes tasks to client modules based on the batch workload.

This section provides testing methodology that describes the tests used to perform the InputAccel Server benchmark testing, server benchmark results, and recommendations to tune the InputAccel Server to maximize performance. The results help determine the typical task processing rate of the InputAccel Server, the Central Processing Unit (CPU) utilization based on the number of active batches in memory, and provide sizing recommendations that determine the number of InputAccel Servers or CPU cores needed.

Benchmark results may vary for your specific production environment. Use the results and recommendations presented in this section as a starting point to help you determine the appropriate sizing of the InputAccel Server for your specific production environment and to tune the InputAccel Server to optimize its performance.

## Test Environment and Methodology

This section describes the testing methodology used during the benchmark testing of the InputAccel Server.

### CaptureFlow Used for InputAccel Server Testing

A benchmarking CaptureFlow process was created with CaptureFlow Designer that represented a typical set of processing steps, as describd in the following table. Most steps included some server-side scripting.

**Table 1. Processing Steps in CaptureFlow Process Used for IA Server**

| Module Name | Trigger Level | # Tasks Generated per Batch | Comments |
|---|---|---|---|
| Standard Import | 7 | 1 | Imports 50 files (multipage TIFF) at document level |
| IABatchCreationMerge (SYNC) | 7 | 1 | Standard XPP sync step |
| .NET Code Module | 7 | 1 | Used to get configuration parameters |
| Multi | 7 | 1 | Dummy step (do nothing) |
| Multi | 7 | 1 | Does nothing (deletes batch in case of 1 page only) |
| ODBC Export | 7 | 1 | Export early statistics |
| Image Converter | 1 | 50 | Split document level files into pages (4 pages each) |
| Image Processor | 0 | 200 | Minimal filters applied |
| Multi | 7 | 1 | Split on barcode, delete blank pages (none exist) |
| Multi | 7 | 1 | Delete empty documents (none exist) |
| Multi | 7 | 1 | Delete empty folders (none exist) |
| Classification | 2 | 1 | Assigns template codes, may restructure batch |
| Captiva Identification | 2 | 1 | No user work done – just receives task and finishes task |
| Extraction | 1 | 50 | Extract OCR zones |
| Captiva Completion | 1 | 50 | No user work done – just receives task and finishes task |
| NuanceOCR | 0 | 200 | Create PDF “Image-only” for each page |
| Image Converter | 1 | 50 | Assemble PDF pages into multi-page PDF at document level |
| Standard Export | 1 | 50 | Export document data to CSV file and images (as multi-page PDF) to file system |
| .NET Code Module | 7 | 1 | “Do nothing” script, placeholder for future use |

| Module Name | Trigger Level | # Tasks Generated per Batch | Comments |
|---|---|---|---|
| ODBC Export | 7 | 1 | Update statistics about batch |
| SYNC (checks age of batch) | 7 | 1 | “Do nothing” step |
| Enter Decision Block | 7 | 1 | Check age of batch – determines batch is old enough and |
| Exit Decision Block | 7 | 1 | |
| ODBC Export | 7 | 1 | Final update of statistics |
| IADone | 7 | 1 | Standard XPP step just prior to batch end. |
| Multi (delete batch) | 7 | 1 | Delete batch |
| TOTAL TASK COUNT | | 669 TASKS | |

# Method of Testing

### Maintaining the batch load:

A total of 5 instances of Standard Import were used to monitor 5 different folders at the same time, allowing for new batches to be created as quickly as the InputAccel Server would allow as long as there were always new files to be found in the import folders. The end effect was that in each directory poll, 300 files were seen in each directory and resulted in the creation of 6 new batches (50 files per batch), after which each module instance would wait 6 seconds and poll the folder again, at which point it would discover a new set of 300 files and the cycle would repeat as long as there were files to be found in the folders.

An external Windows VB Script continually monitored the total batch count on the InputAccel Server by querying a performance counter value. When the number of batches was found to be *below* a certain minimum threshold, the script would repopulate the import folders being monitored by Standard Import so that batch creation would continue. If the number of batches was found to be *above* a maximum threshold, the script would *not* repopulate the import folders and would instead sleep for a short period and then resume checking the total batch count on the InputAccel Server. The result was that the batch count would eventually drop below the threshold as batches were completed and deleted, and that would in turn trigger the script to resume creating new batches until the maximum threshold was reached again.

Through this technique, an empty InputAccel Server would first build up a set of batches by creating them faster than they could be completed. Once the threshold was reached, it would maintain that specific batch count indefinitely. The rate of new batch creation would continually "self-throttle" to always match whatever was the current rate of batch completion.

One of the key differences in this approach compared to earlier server testing approaches is the maintaining of a large number of active batches on the server. In most of the new tests in this guide, the threshold was set to maintain 2,500 batches on the server at all times (both minimum and maximum threshold values were set to 2,500). This is a more realistic user scenario than was used in earlier versions of this guide and puts more strain on memory resources as compared with the tests from earlier versions of this guide in which only a small number of batches existed on the server at any given time.

### Behavior of the Batches:

Each batch initially consisted of 50 document-level files which were multipage TIFF images imported from a file system by Standard Import. After splitting these into pages using Image Converter, the result was 50 document nodes with 4 pages each. The 1st page was a special invoice image which could be classified as such, and data extracted using zonal OCR. The remaining 3 pages were simply classified as "attachments" without any OCR extraction requirements.

The 200 page batch was next processed through Image Processor, Classification, Identification, Extraction, and Completion. After this, each page was converted into a PDF using NuanceOCR at page-level and then these pages were combined into a multipage PDF file with Image Converter and then exported to file system with Standard Export in the form of 2 files per document: a CSV file with minimal document level data values and the multipage PDF file containing 4 pages. Once exported, the batches would be automatically deleted.

At 3 points during the processing, some statistics and timing information about each batch were exported to a database for later analysis using ODBC Export. At 2 points in the batch, the .NET Code Module executed some custom code. During all processing steps, a small amount of CaptureFlow scripting ran on the server to capture timestamps in the step "Prepare" and "Finish" events.

Identification and Completion were automated through internal mechanisms to make them automatically accept tasks.

### Timer Rule Disruptions:

During the benchmark tests, batches would normally be processed in a "first in, first out" sequence based on the batch creation date. When all batch priorities are equivalent, the InputAccel Server sends tasks from the oldest batches first. This can lead to a very predictable and uninteresting sequential processing of batches.

In a real environment, batches may be processed in a less predictable order due to users selecting specific batches in Identification and Completion or due to changes in batch priorities based on business logic.

In order to cause some intentional "disruption" in the sequence of batch processing, a rule was defined for the IA Timer module. This rule would run every 10 minutes and would change batch priorities, cycling through 3 different priority values. The logic was like this:

If Priority=50 change to 20
If Priority=40 change to 50
If Priority=30 change to 40
If Priority=20 change to 30

All 4 of these rules would execute in sequence at the 10 minute interval. The result would be something like this:

- Batch originally has priority 50
- Within 10 minutes, it is changed to 20, but soon thereafter changed again to 30
- After 10 minutes it is changed to 40
- After 10 minutes it is changed to 50
- After 10 minutes it is changed (briefly to 20) and then again to 30

At any given time, the batches will typically have any one of the 3 priorities of 50, 40, or 30. A batch will only rarely and for a very short time have a priority of 20.

In cases where not all batches could fit into the virtual memory, the server would need to unload some batches and load other batches when the priorities changed.

This Timer Rule was not used in all tests. The benchmark results for each test indicate whether this Timer Rule was used or not.

## Test Environment

### Client Module Environment:

To process so many batches as quickly as possible, many copies of client modules were launched on an array of virtual machines. Some VMs were very low powered (2 CPU), some had moderate power (8 CPUs), and a few had very high power (24 CPUs). The VMs were intentionally "overloaded" with a mix of client modules to ensure that the CPU capacity of the VMs was heavily utilized and to minimize times when a VM may be underutilized because some modules had no work at a given moment.

In all, a total of 510 client module connections were used, as described below:

| Module Name | # of Connections |
|---|---|
| Standard Import | 5 |
| .NET Code Module | 6 |
| Captiva Completion | 92 |
| Standard Export | 12 |
| Extraction | 94 |
| Image Processor | 98 |
| Classification | 32 |
| Captiva Identification | 20 |
| Multi | 6 |
| ODBC Export | 6 |
| Image Converter | 46 |
| NuanceOCR | 92 |
| Timer | 1 |

Note that due to overloading the VMs, the above client modules were not necessarily running at their maximum ideal throughput, but the sheer numbers of connections offered some compensation to offset that.

### Hardware Environment for InputAccel Server:

Refer to Machine Configuration Used for InputAccel Server: 64-bit Improvements for details on the hardware configuration and virtual machine settings used for 64-bit InputAccel Server tests.

## InputAccel Server Performance Results

InputAccel Server performance is measured in terms of “tasks per hour”.

While the size and nature of a task varies with the type of client module used and the trigger level of the task, every task has some impact on server performance. Therefore, calculate the rate at which tasks are processed by the InputAccel Server before estimating the number of InputAccel Servers required.

The number and rate of tasks processed by the InputAccel Server in any time period depends on the following key factors:

- The design of the process (how many processing steps, which trigger levels, etc.).
- The structure of the batches (how many documents, how many pages per document, etc.).
- The business logic embedded in the process that dictates when a module step should be triggered.
- The time period in which all tasks are expected to be processed.

The results of InputAccel Server testing yielded the following benchmark results:

### SCENARIO 1: IMPACT OF LARGER VIRTUAL ADDRESS SPACE

Background:

One of the key advantages of converting the InputAccel Server version 7.5 to run as a 64-bit process is that it can utilize all the memory in the system for batch processing.

All earlier versions of the server were 32-bit and therefore limited to a total maximum process address space of 4 GB. Since batches must be mapped into memory to be processed, and only a limited number of batches could fit into memory at any given time, the 32-bit InputAccel Server would frequently unload some batches from memory in order to load other batches into memory. This frequent loading and unloading resulted in significant disk overhead and batch processing interruptions which slowed the overall processing rate of the server.

With a 64-bit server all batches can be retained in memory at all times and therefore batches never need to be swapped in our out of memory. This eliminates the phenomenon of “batch thrashing” which was a performance problem in the past.

Test description:

The following table contains 2 sets of test results which highlight the benefits of retaining all batches in memory.

Both sets of results compare batch processing rates when batches are processed sequentially, in a First-In, First-Out manner (FIFO), and again when batches are not processed in a consistent order (Non-FIFO).

When batches are processed in FIFO order, the batches are completed, exported, and deleted in precisely the same sequence as they were created. Even in a constrained memory configuration, this type of processing results in a relatively low number of batch loads and unloads, and these are performed at uniformly distributed intervals such that their impact is minimal.

In a Non-FIFO sequence, batches must be more frequently loaded and unloaded into memory at inconsistent intervals and often in larger numbers all at once. The Non-FIFO sequence was achieved by using the IA Timer module to execute a rule on a 10 minute interval which would change all the batch priorities, resulting in the server having to begin or resume processing an entirely different set of batches. The consequence of this in a constrained memory configuration is that many batches would need to be unloaded from memory so that other batches could be loaded into memory.

The test results on the next page show the effect of FIFO vs Non-FIFO processing first in a constrained memory configuration, and then again in an unconstrained memory configuration where all batches were in memory at all times.

**Table 2. Benefit of larger virtual address space with Non-FIFO batch processing**

| Scenario | Tasks / Hour | Tasks / Hour (per 100% CPU Core)* | Memory configuration Total RAM / BMASK setting | Batches in virtual memory / total batches | Virtual Bytes Used by ias64 process | Batch Loads/sec (perf cntr) | Total CPU Utiliza-tion (on 24-core) | CPU Utilization distribution (averaged) |
|---|---|---|---|---|---|---|---|---|
| Constrained Batch Memory | | | | | | | | |
| (A)7.5 XPP **FIFO** | 1.79 M | 149 K | 230 GB **3 GB** | 313 / 2500 | 3.0 GB | 2.1 | 21% | 12 @ 37% 12 @ 5% |
| (B)7.5 XPP **NON-FIFO** | 1.54 M *(-14%)* | 128 K | 230 GB **3 GB** | 349 / 2500 | 3.0 GB | 7.3 | 18% | 12 @ 32% 12 @ 5% |
| Unconstrained Batch Memory | | | | | | | | |
| (C2)7.5 XPP **FIFO** | 1.79 M | 149 K | 230 GB **220 GB** | 2500 / 2500 | 8.3 GB | 0.7 | 21% | 13 @ 31% 11 @ 9% |
| (D)7.5 XPP **NON-FIFO** | 1.61 M *(-4%)* | 134 K | 230 GB **220 GB** | 2500 / 2500 | 5.7 GB | 0.7 | 19% | 13 @ 28% 11 @ 9% |

* Tasks/Hour rate divided by 12, since in most cases there were 12 active CPUs (the other 12 were hardly utilized)

Example: 1.79 million / 12 = 149 thousand

Conclusions:

- With constrained memory (3 GB), the batch processing rate dropped 14% in this example due to heavier batch swapping, which can be seen in the performance counter value **Batch Loads/sec**.
- With unconstrained memory, the batch processing rate dropped only about 4% and the performance counter **Batch Loads/sec** showed no difference, as would be expected
- While this was an artificial example, in a real-world scenario with constrained memory there could be frequent batch swapping due to
    - Users picking work per department, resulting in batches being processed in a different order than they were created
    - More extensive IA Timer rules which attempt to read or write IA values in all batches, causing heavy periods of batch loading/unloading
- With unconstrained memory, all batches are available for processing at all times and never need to be swapped

Suggestions for configuring memory:

Ensure that there is sufficient memory in the system to accommodate all batches in memory, and configure the server parameter **BatchMaxAddressSpaceK** to be about ½ GB less than the total memory in the system. This requires making an estimate of the total memory required for all batches. Use the following guidelines as an example:

1) Determine the maximum size that an IAB file (batch data file) might attain in your environment. This can be done by
    a. Processing some batches of a typical size through all steps but not deleting them
    b. Pause the InputAccel Server service, and then Resume it again -- this will possibly make the IAB files grow a little more
    c. Look through the InputAccel Server's "batches" folder for the largest IAB file size
2) Estimate the maximum (peak) number of batches that may ever reside on the server at any given time
3) Multiply the IAB file size by the maximum number of batches to get the total disk space required for these IAB files, then convert this into GB
4) Allocate an additional 8 GB *or more* RAM above and beyond the maximum value for batches to the system as more RAM can help Windows operate more efficiently and with less disk access

Example:

1) You estimate your IAB files will reach 25 MB in size when finished
2) You expect to normally have about 2,000 batches on the system in production, but on some rare instances you may have as many as 3,000
3) 25 MB x 3,000 = 75,000 MB = 75 GB (let's divide by 1,000 rather than 1,024)

4) 75 GB + 8 GB = 83 GB

    a. For a virtual machine, allocate at least 83 GB RAM

    b. For a physical machine, you are constrained to memory module sizes, so more likely the system would have 96 GB or 128 GB RAM

5) Subtract about ½ to 1GB from this to get the BatchMaxAddressSpaceK setting -- note that you must *accurately* convert it into KB for the server setting

    a. 83 GB – 1 GB = 82 GB

    b. 82 GB * 1024 * 1024 = 85,983,232 KB

    c. Set BatchMaxAddressSpaceK = 85983232.

    d. Confirm this value is correctly set by viewing the debug.out file on the server after a restart. The hex equivalent of this would be 0x5200000.

## SCENARIO 2: IMPACT OF MULTITHREADED WORKFLOW PROCESSING

Background:

For many versions now, the InputAccel Server has supported multithreaded client I/O (ability to interact with multiple client modules at the same time) but until version 7.5 there was always a single workflow engine (VBA) which processed all of the CaptureFlow logic and calls to CaptureFlow scripting.

This VBA engine in 7.1 (and earlier) was 32-bit and ran "in-proc" within the InputAccel server process. Beginning with the 7.5 InputAccel Server, there have been 3 significant changes relating to the workflow engine processor:

1) XPP-based CaptureFlows are now compiled into *native .NET code* rather than interpreted VBA code, and the server now includes a corresponding engine to execute the .NET-based XPP CaptureFlows in-proc under the .NET Framework

2) VBA-based CaptureFlows* continue to execute within a 32-bit VBA engine, but this engine now runs *"out of proc"* from the server process due to the InputAccel Server being 64-bit

3) Both types of workflow engines (.NET and VBA) support multithreading, allowing the CaptureFlow code of more than one batch to be executed at the same time

*Note that "VBA-based CaptureFlows" includes both

- IPPs of *any* version (including 7.5 Process Developer)

- XPPs developed in any version of CaptureFlow Designer prior to 7.5 and not yet recompiled under the 7.5 version

These changes in the 7.5 InputAccel Server raise several possible questions:

- How does an existing VBA-based process perform under 7.5 now that it's running out of proc?

- How will an older XPP perform under 7.5 after it's been recompiled into .NET?
- How can I optimize the performance through configuration of multiple VBA or VBNet threads on the server?

Test description:

The test results in the following table will help to address these questions. All of these tests were performed under similar conditions:

- 230 GB system RAM
- BatchMaxAddressSpaceK configured to 220 GB
- 24 virtual CPUs (in a 2-node NUMA virtual architecture)
- All batch processing was straight-through FIFO
- Identical XPP
    - Compiled under 7.1 for testing of VBA CaptureFlows
    - Compiled under 7.5 for testing of VBNet CaptureFlows

Tests were run comparing

- 7.1 XPP on 7.1 Server
- 7.1 XPP on 7.5 Server
- 7.5 XPP (recompiled) on 7.5 Server
- Various numbers of threads configured for the workflow engine

Table 3. 3-way comparison of XPP performance (7.1 and 7.5 XPP versions)

| Scenario | Tasks / Hour | Tasks / Hour (per 100% CPU Core)* | Max VBA or VBNET threads | Batches in virtual memory / total batches | Virtual Bytes Used by ias64 process | Batch Loads/sec (perf cntr) | Total CPU Utiliza-tion (on 24-core) | CPU Utilization distribution (averaged) |
|---|---|---|---|---|---|---|---|---|
| **VBA on 7.1 Server** | | | | | | | | |
| 7.1 XPP on 7.1<br>**1 VBA In-Proc** | 1.3 M | 107 K | n/a | 227 / 2500 | 3.0 | 1.6 | 12% | 12 @ 22%<br>12 @ 2% |
| **VBA on 7.5 Server** | | | | | | | | |
| 7.1 XPP on 7.5<br>**3 VBA Threads** | 0.86 M | 71 K | 3 | 2500 / 2500 | 17.1 | 0.4 | 25% | 3 @ 80%<br>21 @ 17% |
| 7.1 XPP on 7.5<br>**12 VBA Threads** | 1.09 M | 91 K | 12 | 2500 / 2500 | 16.8 | 0.5 | 54% | 17 @ 65%<br>7 @ 27% |
| **VB .NET on 7.5 Server (same XPP recompiled)** | | | | | | | | |
| 7.5 XPP on 7.5<br>**3 VBNET Threads** | 1.86 M | 155 K | 3 | 2500 / 2500 | 5.3 | 0.8 | 21% | 4 @ 50%<br>20 @ 15% |

* Tasks/Hour rate divided by 12

Conclusions from the 3-way comparison:

### IPP-based processes

- IPP-based processes (VBA) running on the 7.5 InputAccel Server may execute a little slower and consume a little more CPU resource than under previous versions of the server
- The degree to which this would be noticed depends on several factors such as
  - Degree of load on the InputAccel Server
  - Complexity of the IPP code (particularly the number of IA value reads/writes)
  - Number of total batches on the InputAccel Server
- Some factors which may offset the VBA performance differences include
  - Configuring **MaxVBAHost** server parameter
  - Providing sufficient RAM to hold all batches in memory
    - In cases where all batches could not fit into memory and were frequently loaded/unloaded, the performance improvement of unconstrained 64-bit memory access may offset the small decrease in VBA code execution
  - Upgrading the hardware with faster CPUs and/or memory

### XPP-based processes

- Recompiled XPPs will execute *faster* on the 7.5 server than they did on previous server versions running under VBA
- For optimal performance, XPPs should be recompiled in the 7.5 version of Captiva Designer and redeployed to the server
  - The XPP file will need to be saved under a new name
  - The deployed process names within the XPP will be retained
- It is recommended to complete all batches on the server based on this process prior to upgrading the process if the deployed process name remains the same

### Optimal configuration of VBA and VBNET threads

- See the results in the following 2 tables for guidelines on configuring thread counts with server parameters MaxVBAHost and MaxVBNetHost

**Table 4. Comparison of MaxVBNetHost settings (for 7.5 XPPs)**

| Scenario | Tasks/ Hour | Processing Rate per CPU Core* | Max VBNet Threads | # Batches in virtual memory / total batches | Virtual Bytes Used by ias64 process | Batch Loads/sec (perf ctr) | Total CPU Utilization (on 24-core) | General CPU utilization distribution |
|---|---|---|---|---|---|---|---|---|
| (E)7.5 XPP<br>**3 THREADS** | 1.86 M | 155 K | 3 | 2500 / 2500 | 5.3 GB | 0.8 | 21% | 4 @ 50%<br>20 @ 15% |
| (G)7.5 XPP<br>**6 THREADS** | 1.85 M | 154 K | 6 | 2500 / 2500 | 5.4 GB | 0.8 | 21% | 12 @ 38%<br>12 @ 4% |
| (C2)7.5 XPP<br>**12 THREADS** | 1.79 M | 149 K | 12 | 2500 / 2500 | 8.3 GB | 0.7 | 21% | 13 @ 31%<br>11 @ 9% |
| (F)7.5 XPP<br>**24 THREADS** | 1.60 M | 133 K | 24 | 2500 / 2500 | 6.1 GB | 0.7 | 19% | 12 @ 32%<br>12 @ 5% |

* Tasks/Hour rate divided by 12, since in most cases there were 12 active CPUs (the other 12 were hardly utilized)

Conclusions:

- Among these test results, the best performance was achieved with 3 VBNet threads
- Adding additional VBNet threads beyond 3 actually began to hurt performance
- General recommendation would be to initially set VBNet threads to about ¼ of the total # of CPUs
- You could experiment with increasing this value, however adjusting it any higher than ½ of total # of CPUs will most likely hurt rather than help
- In these tests, the virtual machine actually had 24 CPUs allocated, but was predominately using only 12 of them for InputAccel due to NUMA architecture and "local" memory access of one of the NUMA nodes. For this reason, we are considering this to be a 12 CPU system in terms of recommending ¼ to ½ CPU count for the **MaxVBNetHost** server setting.

**Table 5. Comparison of MaxVBAHost settings (for IPPs or older XPPs)**

| Scenario | Tasks/ Hour | Processing Rate per CPU Core* | Max VBA Threads | # Batches in virtual memory / total batches | Virtual Bytes Used by ias64 process | Batch Loads/sec (perf ctr) | Total CPU Utilization (on 24-core) | General CPU utilization distribution |
|---|---|---|---|---|---|---|---|---|
| 7.1 XPP<br>7.5 IA Server<br>**3 THREADS** | 0.86 M | 143 K | 3 | 2500 / 2500 | 17.1 | 0.4 | 25% | 3 @ 80%<br>21 @ 17% |
| 7.1 XPP<br>7.5 IA Server<br>**12 THREADS** | 1.09 M | 84 K | 12 | 2500 / 2500 | 16.8 | 0.5 | 54% | 17 @ 65%<br>7 @ 27% |
| 7.1 XPP<br>7.5 IA Server<br>**24 THREADS** | 1.11 M | 71 K | 24 | 2500 / 2500 | 16.8 | 0.5 | 65% | 12 @ 69%<br>12 @ 61% |

* Tasks/Hour rate divided by 12

<u>Conclusions:</u>

- Among these test results, the best performance was achieved with greatest number of VBA threads (24) however the increase was marginal and came at the cost of significantly increased CPU utilization
- Although not specifically tested here, 6 threads is probably the ideal number in terms of tradeoff between performance and CPU utilization
- General recommendation would be the same as for the VBNet thread count – set **MaxVBAHost** to a value between 3 and 6 and or about ¼ to ½ of your CPU count
- It is also advisable to monitor the CPU utilization during high volume production and if the utilization becomes excessive (say, sustained rate of over 50%) then consider lowering the **MaxVBAHost** count slightly.

## SCENARIO 3: IMPACT OF THE USE OF REPORTING LOG RULES

Background:

Enabling the log rules for reporting will have an impact on performance. The degree of impact depends on several factors such as

- The size of the reporting tables
- The speed of the SQL Server
- The real-time processing rates of all InputAccel servers which are sharing the same database (such as in a ScaleServer group)

As the reporting tables grow, the stored procedures which update the tables can take longer to execute on the SQL server.

The InputAccel Server will cache large queues of transactions for the log rules in memory so as to not impact server performance. However if these queues eventually fill up completely, the InputAccel Server will suspend batch processing for a very brief moment to allow the queues to reduce slightly. This sort of behavior can lead to the server processing rate becoming directly tied to the stored procedure execution rate.

The following test results were made with a slightly different XPP than in the previous tests, so these results are not directly comparable with previous results. However, the results below are useful to see the impact that reporting can have as the tables grow. The log rules which were enabled were

- ReportBatchCreate
- ReportBatchDelete
- ReportBatchRename
- ReportNodeCreate
- ReportNodeDelete
- ReportTaskFinishCreatePage
- ReportTaskFinishDonePage
- ReportTaskFinishIndexTask
- ReportTaskFinishTask

**Table 6. Impact of Reporting Log Rules**

| Scenario | Tasks/ Hour | Processing Rate per CPU Core* | Max VBNet Threads | # Batches in virtual memory / total batches | Virtual Bytes Used by ias64 process | Batch Loads/sec (perf ctr) | Total CPU Utilization (on 24-core) | General CPU utilization distribution |
|---|---|---|---|---|---|---|---|---|
| Initial Rate on day 1 (similar to not having log rules) | 1.53 M | 128 K | 12 | 2500 / 2500 | 22.9 GB | 0.6 | 25% | 12 @ 40%<br>12 @ 11% |
| Average rate on day 2 | 1.18 M | 99 K | 12 | 2500 / 2500 | 30.8 GB | 0.5 | 20% | 12 @ 32%<br>12 @ 7% |
| Average rate for days 3-8 where tables have grown | 0.86 M | 72 K | 12 | 2500 / 2500 | 28.1 GB | 0.3 | 14% | 12 @ 25%<br>12 @ 3% |

* Tasks/Hour rate divided by 12, since in most cases there were 12 active CPUs (the other 12 were hardly utilized)

<u>Conclusions:</u>

- Depending on speed of SQL Server, processing rates under heavy continuous load can reduce potentially by as much as 40% when reporting is used
- Careful tuning of SQL Server is recommended to minimize this impact

# InputAccel Server Sizing Recommendations

To determine the number of InputAccel Servers required, or the number of CPU cores required, use the following approach:

1. Use the Captiva Capture Batch and Process Modeler (Batch and Process worksheet) to estimate the total number of tasks you will be processing in one day.

   **Example**: Your volume of 500,000 pages per day equates to (based on your process design) 10,000,000 tasks per day.

2. Divide the daily number of tasks by the number of production hours in a typical day to get the "tasks per hour" rate required by your environment.

   **Example**: Your work day is 10 hours long, so you need to process 1,000,000 tasks per hour.

3. Divide your required "tasks per hour" rate by 100,000 "tasks per hour per CPU" (the lower end of performance rate seen in internal testing, normalized to a single CPU running at about 30% utilization).

   **Example**: 1,000,000 / 100,000 = 10 CPU cores

4. Pad the result by adding about 50% more (multiply by 1.5) to the number of cores. This will help cover spikes in processing levels that are higher than you calculated.

   **Example**: 10 CPU cores * 1.5 = 15 CPU cores

5. If you can find a single server-class machine with sufficient CPU cores ***per socket***, that is all that is required. For example, some high-end Xeon processors have 15 or more cores on a single CPU socket.

   **NOTE about NUMA architecture:**

   Dual-socket systems with high CPU core counts typically use NUMA architecture and divide the memory equally between the 2 sockets. One socket has fast access to 50% of the RAM, and the other socket has fast access to the other 50% of the RAM. This is called "local" memory and is very fast. These sockets can access the "foreign" memory connected to the other socket, but this sort of access is slower. Windows will intentionally prioritize threads onto CPU sockets which have the fastest access to the RAM. Because of this, Windows can end up assigning all InputAccel Server threads to the cores in just one of the sockets, and therefore not really take advantage of the other CPU cores in the other socket. In such NUMA architectures, you should consider only the cores available in one of the sockets when determining how many cores are actually usable by the InputAccel Server.

6. If you are very close to what a single CPU socket can deliver (for example you have processor with 12 cores) you might be sufficiently covered, since these numbers are conservative. Only live performance testing can say for sure.

7. If you definitely need more CPU sockets, because you have for example only systems with dual 8-core CPUs, you may need to configure 2 InputAccel servers in a ScaleServer group. This is particularly true if the dual 8-core system uses NUMA architecture. If you have a 16 core system based on SMP architecture where all CPUs have equivalent access to RAM on a shared memory bus, then this system will fully utilize all 16 cores and would be a good candidate for this example scenario.

   **Example**: Your calculations lead you to the conclusion that you need 15 CPU cores. Since you don't have access to a machine with 15+ cores per socket, you could use two machines –each with a minimum of eight cores in a single CPU socket.

**Note:** Using two machines requires purchasing two InputAccel Server licenses (greater cost), but will also provide improved performance because each InputAccel Server machine will have its own memory and disk resources. All the client/server transactions will be divided between the two InputAccel Servers. The two servers could be configured as a ScaleServer group to make them appear like a single server to the client modules.

# InputAccel Server Tuning Recommendations

The following section provides recommendations about different deployment considerations that can affect InputAccel Server performance.

## CPU

The InputAccel Server uses multi-threaded client Input/Output (I/O) to communicate with client modules. This means that the more CPU cores the InputAccel Server has available to it, the more client connections it can interact with simultaneously. Additionally, activities such as logging data to the database are performed on a separate thread, and therefore can be processed in parallel with the client I/O. Multiple CPU cores are very important to obtain optimal InputAccel Server performance.

- Beginning with version 7.5, we now recommend 12 CPU cores (per socket) for a production system. A lesser system (perhaps 6 cores) is acceptable for testing and development purposes.
- Do not count the additional CPUs exposed through hyperthreading as real CPUs. Count only true number of physical CPU cores in a single socket.

## Memory

Now that the InputAccel Server can access practically unlimited memory, provide sufficient RAM to your system to allow all batches to remain in memory. Refer to the earlier test results for the scenario "SCENARIO 1: IMPACT OF LARGER VIRTUAL ADDRESS SPACE" for details on memory recommendations.

## Disk

A high performance disk array is another key element to improve the InputAccel Server performance. The InputAccel Server makes extensive use of the disk subsystem, as it must manage all of the data, images, and other files required by the client modules to process their tasks.

If the disk subsystem is slow, the InputAccel Server's overall performance level will decrease. One common consequence of a slow disk system is frequent "Waiting for response from server" messages on the client machines.

To improve InputAccel Server performance, especially under heavy processing loads, it is strongly recommended to use a high-performance disk array that includes a hardware caching controller (with both read and write cache enabled) and fast disk drives, 15000 Revolutions Per Minute (RPM), for example.

While RAID 5 and 6 are popular configurations for fault tolerance due to their more economical use of disks, they can suffer from performance issues due to the additional overhead of calculating and storing redundant data. A faster but equally fault tolerant configuration is RAID 10.

**Recommended Disks for Best Performance:**

- Hardware-based RAID 10 disk array

- Caching controller (with battery backup) configured for:
  - Write back caching
  - Adaptive read-ahead caching

**Recommendations When Using SAN or NAS Devices:**

- Storage Area Network (SAN) configured for high performance that is comparable to a local RAID 10 array.
- Network-Attached Storage (NAS) is not recommended because it can have network issues and possible outages.

**Additional Tips to Improve Disk Performance:**

- **Virus scanning**: Exclude the entire \IAS folder (InputAccel Server's data folder) from on access virus scanning. As an alternative, schedule a full virus scan of the \IAS folder to occur during off-production hours when the InputAccel Server service can be paused or stopped.
- **Location of \IASfolder**: Locate this folder on a dedicated disk drive or array so that the drive is not shared with other applications that make heavy use of the disk, such as database servers or the Windows swap file.

# Network

A 1 GB Ethernet connection is sufficient for most applications, however low latency between the clients and servers, or between servers in a ScaleServer group is very important. High latency connections such as in a Wide Area Network (WAN) should be avoided, as this can negatively affect performance of client modules. This is most critical for the attended modules where an operator is interacting with the system and should not have to experience delays in retrieving or saving data.

# Database

When used with a SQL database, the InputAccel Server must have fast and uninterrupted access to the database server. Therefore, it is recommended that the InputAccel Server is located on the same Local Area Network (LAN) as the database server. A remote database server on a Wide Area Network (WAN) connection would be detrimental to InputAccel Server performance. Refer to *InputAccel Database over WAN* for additional information.

If the Reporting feature is enabled and used, anticipate up to 40% reduction in InputAccel Server throughput for a sustained high volume scenarios. For lighter throughput levels or with very fast SQL Server systems, the amount of reduction may be less.

**InputAccel Server Configuration Parameters**

The following sever parameters can affect performance. Modify these settings on the **Server Settings** pane in Captiva Administrator.

- *BatchMaxAddressSpaceK*(default = 3145728)

  Specifies how much virtual address space the server can use. Set this parameter to ½ to 1 GB less than the total system RAM. Note that this parameter is given in KB, so you must convert GB to KB before entering this parameter.

- *BatchMaxLoaded*(default = 100000000)

Specifies how many batches can be loaded into memory at any given time. Set this value to an extremely high number that will never really be achieved, such as 100,000,000.

- *BatchSync*(default = 300)

  Specifies how often the server should sync or commit the batch in memory back to disk. It is like making a snapshot of the batch. If the server should crash, all the batches are restored to their previous snapshot state on disk. The default value is 300 seconds (five minutes). Increase this value to reduce some disk activity which could be beneficial if the disk becomes a bottleneck.

  However, increasing this value also puts more data at risk if the server crashes, because the snapshots could be old enough as to not contain any recently added data. In general, do not alter this value unless you have a specific reason or recommendation to do so.

- *BatchSyncMaxTime*(default = 0)

  Specifies the maximum amount of time (in seconds) that the server can spend performing sync operations on batches in need of a batch sync. The default is 0, which means there is no limit. Normally this is sufficient, but since the InputAccel Server can now keep many more batches in memory at one time, there is a chance that batch sync operations may take longer because more batches are being synchronized at once. You may consider setting some maximum value here, such as 20 or 30 to limit the time spent on each batch sync operation .

- *FileTraceLevel*(default = 116)

  A diagnostic value that specifies the level of information to be written to the InputAccel Server's `debug.out` file. It can be adjusted to have the InputAccel Server output more low-level detailed information about its activities, but this extra level detail comes at a cost of additional disk I/O. The normal value for this setting is 116. In general, do not alter this value unless you have a specific reason or recommendation to do so.

# Batch Size

A batch size refers to two things: the size of the IAB file on the server's disk and the number of pages in a batch. The following sections provide details on how a batch size and the number of pages in a batch impact server performance.

## Impact of Large Number of Pages Per Batch

In some older InputAccel Server tests (before the 7.5 version) some conclusions were made about number of pages per batch. These generalizations should still apply to 7.5.

The results for the number of active batches and number of idle batches were performed with 100 pages per batch. These same benchmarks were repeated, but using batch sizes of 1,000 pages, 10,000 pages, and 28,000 pages per batch.

The results show a decrease in throughput as batch size grows:

- **1,000 vs 100 page batches** – throughput dropped about 13%
- **10,000 vs 100 page batches** – throughput dropped about 33%
- **28,000 vs 100 page batches** – throughput dropped about 65%

## Reasons for Degradation

1. Processing flow is impaired more significantly by steps triggered at the batch level, resulting in longer waiting periods of time before work for other steps is released. This leads to some steps having periods where no tasks are processed, then peak periods where large amounts of tasks are processed all at once.
2. When many client modules are working on the same batch at the same time, the batch needs to be locked more often to update the IA values. If client modules process different batches at the same time, the performance is better. Using smaller batch sizes improves the chances of clients getting tasks from different batches, thereby reducing the chance for locking on the batches.

EMC recommends batch sizes of 100 pages for the best InputAccel Server performance. Up to 1000 pages per batch will yield acceptable throughput. More than 1,000 pages per batch is not recommended and may reduce throughput.

## Impact of Small Number of Pages Per Batch

This scenario was not benchmarked. However, many small batches degrades throughput. We recommend batches with 100 pages, but a few batches with 10 pages will yield acceptable throughput. Numerous batches with only 1 page may significantly degrade throughput and is not recommended.

# Recommended Trigger Levels in Processes

The selected trigger level of a module in a process is usually dictated by the business application. In some cases, the module itself dictates the trigger level. For example, Image Processor can only be triggered at level 0.

In other cases, the need to modify the batch structure restricts the use of very low trigger levels. For example, Classification is unable to modify the tree structure (for example, insert document nodes) if it is triggered at level 0 or 1.

In another example, the need to create a multi-page PDF file from a group of pages requires triggering NuanceOCR at level 1.

However, there are cases where you have a choice of trigger level. For example, if you are performing only zonal OCR on individual pages, you can trigger NuanceOCR at any level from 0 to 7 and you will get the same result.

In cases where you have a choice to select the trigger level, consider the following advantages and disadvantages:

### Low Trigger Level (Such as 0): Advantages

- All the work for a single batch can be distributed among different client machines, resulting in faster end-to-end processing of any single batch. This could be useful when there are service level agreements to meet, and each batch must be finished as quickly as possible.
- If there is an error during processing, only a single page may be affected, reducing the time to reprocess it because only the page that failed is reprocessed.

### Low Trigger Level (Such as 0): Disadvantages

- Large numbers of tiny tasks per batch means more work for the InputAccel Server and more time spent by the overhead of each transaction – the client must receive a task, begin processing, and then signal the server that the task is finished. All the overhead involved in the interaction with the server is multiplied by the number of tasks.

- Multi-threaded client I/O means the InputAccel Server can receive simultaneous requests from multiple client modules to update data in the same batch at the same time. When this occurs, the server must fulfill each batch data update request one at a time, which means some clients will wait longer than usual for their data to be updated in the batch. If the clients were triggered at level 7, each module would be exclusively processing a single batch, meaning that the server could process the batch data updates from each client at the same time, because they were all updating different batches.
- The impact of multiple clients updating a the same batch at the same time is compounded when using trigger level 0, as it enables tasks from the same batch to be sent to many different client modules which will all try to update the same batch at the same time.

#### High Trigger Level (Such as 7): Advantages

- A single, large task is less work for the InputAccel Server, as the overhead of sending the task and receiving the result occurs only once per batch, rather than multiple times per page. Level 7 is less overhead for the server.
- Triggering at level 7 results in a single client module updating all the data in that batch. This significantly reduces the negative impact on the batch that can result when multiple client modules attempt to update batch data at the same time.

#### High Trigger Level (Such as 7): Disadvantages

- All the work for a single batch is processed in a single, large task by the client module. If the task is particularly slow, such as an OCR task, it could take a long time before any single batch completes the step. If there are requirements to complete a batch in the shortest possible time, using level 7 is a disadvantage. Additionally, if the backlog of work for the module drops off, the residual work for the modules is not equally distributed, resulting in some modules being idle while other modules continue working on their batches for some time.

  **Note:** While it may seem that level 7 is slower because an individual batch takes longer to complete, in the overall picture, level 7 is actually faster when there is a large backlog of work to process. When there is a large backlog and a large bank of client machines available, the overall processing time for all batches will be somewhat faster at level 7 because each client module will process a different batch simultaneously. So, even though a single batch takes longer to complete at level 7, since many batches are being slowly processed in parallel, the net result is the same, or actually slightly faster at level 7.
- If there is an error during processing, the entire batch must be reprocessed through that step. Recovery from errors can take longer with level 7 tasks for this reason.

## Comparison of IA Server Performance on a Physical vs. Virtual Machine

Benchmark tests compared the performance of an InputAccel Server running on a physical machine and an InputAccel Server running on a virtual machine (under VMware ESXi, 4.1.0.260247), where the underlying hardware and all other components were the same. See *Machine Configuration Used for Server: Virtualized Benchmark Testing* for details physical and virtual machine configurations.

This benchmark showed a 27% reduction in InputAccel Server throughput when using the virtual machine.

**Recommendations when running InputAccel Server on a virtual machine:**

1. Review the recommended InputAccel Server system requirements in the Release Notes. Apply these recommendations to the virtual machine configuration.
2. Verify whether other virtual machines are running on the same machine as the InputAccel Server and therefore sharing physical resources. If physical resources are shared, set resource allocation Reservations to ensure the server meets the recommended system requirements. Without reserving the minimum resource levels, adequate performance of the InputAccel Server is not maintained.
3. Configure the virtual machine with as many virtual processors as the environment will allow. Since virtualized environments do not always support as many virtual CPUs as a physical machine would, a virtualized environment may require more InputAccel Servers to achieve the same total number of CPU cores that a fewer number of physical machines might offer.
4. Use a separate, dedicated drive for the InputAccel Server Data directory (\IAS folder). Do not store InputAccel Server data on the same drive as the operating system. The data drive can be a virtual drive (VMDK file) on local storage or on SAN storage, or a physical SAN drive directly accessible to the virtual machine. Choose the data drive that yields fastest performance.
5. Use the para-virtualized hardware devices in ESX/ESXi when possible. For example, if using a VMDK file for the InputAccel Server data, configure it to use the VMware Paravirtual SCSI adapter (PVSCSI), which may be slightly faster than the LSI Logic SAS adapter. Additionally, use the VMXNET 3 network adapter for optimal performance.
6. Configure Windows so that it does not update the Last Accessed time stamp on each file it reads by executing the following command and then rebooting the InputAccel Server machine: fsutil behavior set DisableLastAccess 1.
7. Install VMware Tools on the guest operating system and enable its clock synchronization feature. Refer to VMware documentation for details.
8. Refer to the white paper Performance Best Practices for VMware vSphere™ 5.1 from VMware regarding additional best practices for performance of virtual machines with ESXi 5.1.

# Chapter 3 InputAccel Database

## Introduction

The InputAccel Database supports several functional areas of an Captiva Capture deployment.

This section describes the different functional data stored in the InputAccel Database, the testing methodology used to test the InputAccel Database, SQL Server system recommendations based on the task volume, and sizing and tuning recommendations for the InputAccel Database. Use the sizing recommendations to help you size the InputAccel Database, and the tuning recommendations to improve the performance of the InputAccel Database.

The sizing recommendations may vary for your specific production environment. Use the recommendations as a starting point to help you determine the appropriate sizing for your specific production environment.

## Data Stored in the InputAccel Database

The InputAccel Database contains data in the following functional areas:

**Table 7. InputAccel Database Data**

| Functional Area | Purpose |
|---|---|
| Configuration Data | Contains global configuration data such as InputAccel Server settings, client module settings (not process-related), license codes, log rules, report definitions, client-side script files, etc.<br>Configuration data are typically read and written infrequently, hense they typically have little impact on overall performance. |
| Work in Progress (WIP) | Contains basic data about each batch and process on the InputAccel Server at any given time. During normal operation, the InputAccel Server continually updates the data in these tables as batches are added or removed from the system, or whenever the task counts or overall status of a batch changes.<br>WIP data is written or updated by the InputAccel Server at moderate intervals that have relatively low impact on the database. While not negligible, this baseline level is low enough to not be of significant concern. The WIP data is read only by the Captiva Administrator web server when information about batches and/or task counts is displayed in the browser, which is infrequent enough to be negligible. |

| Functional Area | Purpose |
|---|---|
| Logs | Contains all Error, Warning, and Audit log entries generated by various components. The contents of this table are displayed in Captiva Administrator.<br><br>Log entries are added only when specific log rules are enabled and a corresponding log event occurs. The frequency of new log entries depends upon the frequency of the events that trigger them. The default logging and audit settings do not generate any significant load on the database.<br><br>The enabling of Audit log rules, however, has the potential to generate large numbers of logs in a short time. Unless needed for compliance reasons, Audits should be disabled during production and only enabled briefly for troubleshooting purposes. |
| Reports | Spanning a number of specialized tables, report data consists of ever growing statistical data about each batch processed by the system. Used to generate reports about past system activity, these data can grow indefinitely unless purge jobs are specifically scheduled in Captiva Administrator.<br><br>Note: The amount of logging for reports depends on which log rules are enabled and the volume of pages processed, and to a lesser extent the nature of the process and the structure of the batch. By default, reporting log rules are disabled. |
| Web Services | Contains data for tracking incoming web service requests to the Web Services Input module. Incoming web service requests and “offline” tasks for this module will create new rows in a table. If the web service request contains an attachment, that attachment will also be stored in the InputAccel Database. As web service requests and batch tasks are completed, the corresponding rows are removed from the table.<br><br>Unless very high volumes of incoming web service requests are simultaneously processed, the impact on database performance from the use of web services is minimal. |

# Test Environment and Methodology

InputAccel Database performance was analyzed using the following techniques:

- Using Windows Performance Monitor to monitor specific SQL Server performance counters during moderate to heavy simulated production-level activity.
- Disabling specific Report type log rules, followed by the processing of significant numbers of batches and the subsequent observation of which tables were updated, the nature of transactions on those tables, and the amount of table growth caused by the transactions. Using this technique, the impact of each Report type log rule was determined.

Refer to *Physical Machine Configuration Used for Database Testing* for configuration used during testing.

## SQL Server Sizing Recommendations

The following recommendations are for the SQL Server hosting only the InputAccel Database:

**Table 8. SQL Server Sizing Based On Captiva Capture Task Volume**

| Captiva Capture Task Volume (Per Hour) | Recommended SQL Server Edition | CPU Core | Disk System |
|---|---|---|---|
| Low (< 50,000) | w/o Reports: Express<br>w/ Reports: Standard | 1–2 | Standard Disks |
| Medium (50,000 - 400,000) | w/o Reports: Express (x64)<br>w/ Reports: Standard (x64) | 2–4 | RAID 5 or 10 with Read/Write (R/W) Caching |
| High (> 400,000) | Enterprise (x64) | 4+ | Hardware RAID 10 with R/W Caching |

**Note:**

- Do not confuse Task Volume with “pages per hour.” Use the *Captiva Capture Batch and Process Modeler* to convert your “pages per hour” rate into “tasks per hour” based on the particulars of your process and batch structure.
- For Low Task Volume, SQL Server Express edition is limited to a maximum database size of 4 GB (or 10 GB with SQL Server 2008 R2 Express). Therefore, it is only viable for hosting the InputAccel Database if you are sure your database will never grow larger than the maximum database size supported by the SQL Server Express editions. If you use Reports or Audits, you will almost certainly exceed this limit; therefore, SQL Server Express could be used only in low volume environments that do not use Reporting or Audit log rules.
- For Medium Task Volume, x64 editions of SQL Server have significant advantages in memory availability and therefore are generally recommended.

## InputAccel Database Sizing Recommendations

The following section provides recommendations to size the InputAccel Database.

### Estimating InputAccel Database Size Based on Database Growth

Use the following table to calculate a very rough estimate of the size of your InputAccel Database.

- If you are not using Reports or Audits, your database size is determined by the Work In Progress usage only.
- If you have Reporting logs enabled (but not Audits), your database size is the sum of the Work In Progress and Reports. Note that the interval at which you purge the Reports tables is critical in controlling the ultimate size of the database.
- If you have Audits enabled, expect the database to grow very quickly and overall InputAccel Server performance to degrade. It is very critical to schedule a periodic purge of the Audit Logs to avoid the InputAccel Database from growing too large too quickly. You should

monitor your database growth over one day of production to estimate how often you will need to purge the Audit Logs.

**Table 9. InputAccel Database Size Based on Data Stored**

| Functional Area | Resulting Database Size |
|---|---|
| **Configuration Data** | Negligible |
| **Work in Progress (WIP)** | 1 Megabyte (MB) x Max Number of Batches on Server<br>**Note:** For the Work in Progress functional area, if the InputAccel Database grows large enough to completely fill the disk on which it resides, the InputAccel Server will be forced to pause until database operations can be restored. |
| **Logs (Errors/Warnings)** | Negligible |
| **Logs (Audits)** | Negligible if audits are not enabled.<br>If all 14 Audit Log Rules are enabled, the database growth will be roughly twice the amount of having all 13 Report Log Rules enabled. Therefore, by estimating database growth for all Report Log Rules, you can get an idea of the growth for all Audit Log Rules (2x all Report Log Rules).<br>Alternatively, you can monitor database growth over one day of production and extrapolate for however many days the system will run before the tables are purged. |

| Functional Area | Resulting Database Size |
|---|---|
| **Reports** (This is a simplified formula.  A more precise number can be created using the **Logging** worksheet in the *Captiva Capture Batch and Process Modeler*) | 0.0005 MB * ((SC0 * PPD) + (SC1 * PPD / PDoc)) * RLR * PID<br>Where:<br>• PPD = Pages per day (daily processing volume)<br>• PDoc = Pages per Document (batch structure)<br>• RLR = number of Report Log Rules enabled (configuration)<br>• SC0 = step count of all level 0 steps in process (process)<br>• SC1 = step count of all level 1 steps in process (process)<br>• PID = Purge Interval Days -- # of days until purge (configuration)<br>• Example: Consider the following scenario:<br>• PPD = 500,000 pages per day (volume)<br>• PDoc = 3.5 pages per document (batch structure)<br>• RLR = 5 Report Log Rules enabled (configuration)<br>• SC0 = 3 level 0 steps (process)<br>• SC1 = 3 level 1 steps (process)<br>• PID = 30 days before each purge (configuration)<br>DB size = 0.0005 MB * ((SC0 * PPD) + (SC1 * PPD / PDoc)) * RLR * PID<br>DB size = 0.0005 MB * ((3 * 500000) + (3 * 500000 / 3.5)) * 5 * 30<br>DB size = 144643 MB or 141 GB<br>The formula for estimating Report Database growth is approximate and intentionally errs on the side of predicting a larger database size than you may actually see in production. Use the *Captiva Capture Batch and Process Modeler* (Logging worksheet) to get a more precise number.<br>**Note:** For the Reports functional area, there are four pre-installed purge definitions included, which help you to purge the Reports and Audit/Error logs.<br>1. Purge Report Summary<br>2. Purge Report Detail<br>3. Purge Audit/Error Logs<br>4. Purge Report Dispatcher<br>*Purge Report Detail* and *Purge Audit/Error Logs* are scheduled to run periodically to keep the size of the tables small, but you will need for verify, using Captiva Administrator, that their schedules meet your business needs.<br>*Purge Report Summary* and *Purge Report Dispatcher* are not scheduled to run. You need to explicitly define a purge schedule for them if required by your business needs.purge definitions included, which help you to purge the Reports and Audit/Error logs. |
| Web Services | Negligible |

## Estimating InputAccel Database Sizing Based on Transactional Impact

The number of database transactions generated by batch processing is directly related to the number of tasks processed by the InputAccel Server. Therefore, to estimate the database transaction rate, it is important to understand the task rate of the InputAccel Server.

Refer to *InputAccel Server Sizing Recommendations* to understand how to calculate the "tasks per hour" rate of the InputAccel Server. Also, refer to *Recommendations for the Environment Used for SQL Server* for system recommendations for the SQL Server based on the InputAccel Server task rates.

After determining the "tasks per hour" rate of the InputAccel Server, estimate the database transactions per hour as follows:

**Table 10. InputAccel Database Transaction Rate**

| Extra Logging Enabled | Estimated Database Transaction Rate |
|---|---|
| None | Database Transactions/hour = InputAccel ServerTasks/hour * 0.075 |
| Reports | Database Transactions/hour = InputAccel Server Tasks/hour * 2.5 |
| Reports + Audits | Database Transactions/hour = InputAccel Server Tasks/hour * 7.5 |
| Audits (no Reports) | Database Transactions/hour = InputAccel Server Tasks/hour * 5.0 |

**Note:** These estimates are based on direct observations of a system operating at a relatively high rate of processing while monitoring the SQL Server database performance counter **Transactions/sec**. These figures should be used as a guidance for computing the number of transactions.

# InputAccel Database Tuning Recommendations

## Defragmenting and Rebuilding Indexes in the InputAccel Database

Depending on the volume and type of processing, periodically defragmenting and rebuilding indexes in the InputAccel Database may help prevent application performance degradation. The InputAccel Database installer installs these two stored procedures into the SQL Server to perform these functions on a small set of pre-selected tables. Use these stored procedures in conjunction with SQL Server scheduled job functionality to defragment and rebuild indexes on a regular basis:

- *up_ReorganizeIndexes:* Defragments all of the indexes in the pre-selected set of InputAccel Database tables. This stored procedure can be run at any time since it does not affect concurrent application query activity.
- *up_RebuildIndexes:* Rebuilds all of the indexes in the pre-selected set of InputAccel Database tables. This stored procedure must be run when application activity is at a

minimum since application queries will not have access to the tables during the rebuilding process.

**Note:**

- These stored procedures can be run manually or as scheduled jobs; however, these stored procedures are not set up to run automatically as scheduled jobs. You can create your own scheduled jobs to run these stored procedures. Refer to the SQL Server documentation for steps to create scheduled jobs.
- Preferably, defragment indexes once a day and rebuild indexes once a week. The most appropriate schedule for these activities depends on your specific implementation, workload and operations schedules, and can only be determined by the customer.

## Purging the InputAccel Database

Because the InputAccel Database may grow very large with all of the report or audit data, it is necessary to periodically purge this data from the system. Purges can be configured and scheduled from Captiva Administrator. Refer to the *Captiva Administration Guide* for instructions.

## Avoid Running Complex Reports

Avoid running reports that use complex queries against large tables, which can put significant time and load on the database. Whenever possible, run reports outside of peak production times, as they can tie up the report tables long enough to begin affecting InputAccel Database performance which will degrade InputAccel Server throughput.

## Store the IA DB Transaction Log and DB Files on Separate Hard Drives

SQL Server writes all transactions to the transaction log. The size and activity of the transaction log can impact the performance of the InputAccel Database negatively in high volume environments. Consider storing the transaction log on a separate drive controlled by a different disk controller for improved performance if you expect heavy usage of the InputAccel Database.

# Chapter 4 Client Modules

## Introduction

Captiva Capture client modules are software modules that perform specific information capture tasks such as scanning images, enhancing images, performing OCR, indexing data, or exporting images and data.

While the InputAccel Server throughput is measured in “tasks per hour”, client module throughput is measured in “pages per hour” or “tasks per hour”. Every client module is measured separately, as each module has different performance characteristics.

A representative set of key client modules using common configurations were selected for benchmarking. The client modules that were benchmarked include:

- Classification
- Extraction
- Image Converter
- Image Processor
- NuanceOCR
- East Euro / APAC OCR
- Advanced OCR/ICR
- General-Use OCR
- Western OCR
- ODBC Export
- Production Auto-Learning
- Standard Export
- Standard Import

**Note:** This chapter covers the benchmark results for *unattended* modules only. To view the benchmark results for attended modules, such as Captiva Completion, ScanPlus, and Captiva Identification, refer to chapter *Components over a WAN Network*.

In addition to these client modules, the Production Auto-Learning - Supervisor component was also benchmarked.

Each client module section provides a brief summary of the module, illustrates the benchmark results for one or more instances of the module, lists the critical factors that impact the module and their significance, and then provides the sizing recommendations for the module.

The benchmark testing results may vary for your specific production environment. Use the results and recommendations presented in this document as a starting point to help you determine the appropriate performance and sizing for your specific production environment.

# Test Environment and Methodology

This section describes the testing methodology used to test and record benchmark results for client modules.

## CaptureFlow Process Used for Testing Client Modules

The CaptureFlow process used to record benchmark results has the following process flow:

ScanPlus(0) -> Route(7) -> <Optional Processing Module> -> <Processing Module> -> ODBC Export(7) -> IADone(7) -> DeleteBatch(7)

The Route step is the Multi client module that parses the process name to determine which processing module to trigger. Only one processing module is triggered, after which the statistics are exported. If a processing module requires that some other module be run before it, then that processing step is triggered first automatically. After that step is done, the process retriggers the Route step again. In the case where a pre-processing step is necessary, the Route step triggers the correct module to be benchmarked when the Route step is run a second time.

## Method of Testing

Performance tests measured the time it took to process a set of identical batches through the module while one, two, or four copies of the module were running as a service.

For every batch, the CaptureFlow process captures the moment in time that the first page of the batch was sent to the client module for processing and the time that the last page was returned to the server from the client module. The CaptureFlow process then calculates the difference and refers to this as the processing duration (for that batch).

For all module benchmarks, only the module throughput was measured. All other components in the environment (such as the InputAccel Server, the network, the SQL Server) were configured to be as fast as possible so they would not be a bottleneck and negatively influence the throughput of the module. This, however, means that the module performance measurements are ideal and probably will not be achievable in a production environment. You will need to scale the benchmark results to your production environment.

For all modules, a benchmark was run using one instance of the module. The results obtained when running a single instance of the module, provided the baseline results. Additional benchmarks were run with two and four instances of the module running on a multi-core machine. The results were compared to the single instance results to show the throughput gained by running additional instances.

A module's "pages per hour" or "tasks per hour" throughput rate was determined by measuring the time taken to process all the pages in all the batches used in the benchmark, then dividing by the total number of processed pages and then scaling to 60 minutes.

To capture the CPU and memory usage of the module while one, two, or four copies were running, the following Windows performance counters were used:

- Disk Read Bytes / Sec
- Disk Writes Bytes/Sec
- Process Private Bytes
- Processor(0)\% Processor Time

- Processor(1)\% Processor Time
- Network Interface Bytes Sent / Sec
- Network interface Bytes Received / Sec
- Processor(_Total)\% Processor Time
- Processor\% Processor Time

# Client-side Tuning Recommendations

Certain older client modules can be configured to optimize performance by setting options in the Settings.ini file. The tuning recommendations in this section apply only to modules which originated from InputAccel 5.3 or 6.x. Other modules such as Image Processor, Image Converter, Captiva Completion, Captiva Identification, Extraction, Standard Import and Standard Export do not use the settings.ini file.

The Settings.ini file is located on the client machine in the %ALLUSERSPROFILE%\EMC\InputAccel folder. By default, these INI parameters are not present in the file, which means the default values are implied. However, if you have explicitly altered these values within a previous version of Captiva Capture and are upgrading, review the Settings.ini file and either remove the parameter (forcing the default value) or verify that the parameter is still correct for your implementation as the new default or optimum value may have changed significantly from previous versions.

**Settings.ini Parameters:**

- *PrefetchDefault*(default = 2)

  This value tells the InputAccel Server how many additional tasks to send to the client module in addition to the task that the server was already planning to send. A few modules explicitly override this setting and do not obey it, but most modules follow this setting.

  To disable prefetching entirely, set this value to 0. Increase this value to prefetch more tasks. One reason to increase it would be to help ensure that a single client module processes most of the tasks from any given batch. Another reason to increase it would be if you have extremely fast processing tasks (e.g. just milliseconds per task). In that case, you can end up with the module waiting for tasks because it is processing them faster than the server can send them.

- *FileCacheSize*(default = 8)

  Determines the number of files cached in local memory. To increase performance, reduce this value when processing large file sizes. To enable scanning of large images (over 50 MB), change this value to 1.

- `CacheSize` (default = 1048576)
  Specifies the size in bytes of the local values cache for IA values and other items. The value range is 8192 and greater. There is not a fixed upper limit.

- *IAClientDebug*(default=0):

  Verify that *IAClientDebug*is not set to 1 (enabled) in the `Settings.ini`. This is sometimes done to capture log files for debugging purposes. When this parameter is enabled, it generates debug logs that can significantly slow down module performance.

- *CacheCount*(default=200,000)

The CacheCount parameter has a larger default value than in previous Captiva Capture versions. Upgrade customers that may have manually modified the CacheCount configuration item in the settings.ini file must remove this setting from the file so the value defaults to 200,000.

Starting with InputAccel 6.5 , modify the CacheCount parameter only if you need to set it higher than 200,000.

# Client Module Recommendations

This section contains test scenarios, test results, and recommendations that have been applied while testing the performance of Captiva Capture client module.

## Classification

Classification performs classification based on graphical templates as well as keywords.

- **Automatic (Standard Templates) Classification**: Graphical based classification using templates created using the automatic learning feature in Recognition Designer.
- **HPA (High Precision Anchor) Classification**: Similar to Automatic (Standard Templates) classification in that it is a graphical based classification, but different in that HPA templates are created manually by placing anchors on a template page and assigning settings to each anchor. Typically, HPA classification is more accurate than Automatic (Standard Templates) one.
- **Text Matching Classification**: Text-based classification. During classification, full-page OCR extracts data from each image and classification matches the document's textual content to the signatures of the text matching templates. This is the slowest method of classification due to the full-page OCR and long text strings used for matching.
- **Keyword Classification**: Text-based classification. OCR is performed, typically full-page, to extract data from each image and classification matches the document's textual content to the keywords and rules defined in the templates. This classification method is much slower than the graphical based Automatic and HPA methods due to OCR.

### Test Scenarios

The following templates and classification methods were benchmarked.

1. 5 templates, Automatic/High Precision Anchor (HPA), Level 7. Project size: 3 MB.
2. 500 templates, Automatic/HPA, Level 7. Project size: 86 MB.
3. 3000 templates, Automatic/HPA, Level 7. Project size: 532 MB.

   Scenarios 1, 2, and 3 only test graphical classification. Full page OCR is not performed. The goal is to measure the impact of the number of such templates on classification speed. Three projects of varying sizes were created to study the impact of speed. These scenarios are not the most CPU-intensive usage of Classification.
4. 5 templates, Keyword Classification, Level 7, five keywords. Project size: 0.5 MB.

   This scenario uses the same batch as scenarios 1-3. It enables a comparison of Full Page OCR classification method against graphical classification methods. The East Euro / APAC OCR, Advanced OCR/ICR, General-Use OCR, and Western OCR engines were used.
5. 100 templates, Text Matching (TM), Level 7. Project size: 14 MB.

   This scenario is the most CPU-intensive classification method for Classification. The East Euro / APAC OCR, Advanced OCR/ICR, General-Use OCR, and Western OCR engines were used.

6. 5 templates (Automatic/HPA), color images, Level 7. Project size: 30 MB.
   This test, when compared against Scenario 1, shows the impact of color images.

Unless otherwise noted, the Western OCR engine was used for all classification benchmarks. The Keyword classification benchmarks in the document performed full-page OCR and used five keyword rules.

## Benchmark Results

Refer to *Test Environment Used for Testing of Classification and* Extraction Modules

System Properties for Capture Client Modules and Administrator

| Item | Required |
|---|---|
| Hardware | Lenovo C30 Workstation<br>• Intel Xeon E5-2609 v2 @ 2.5 GHz (quad core)<br>• 4 virtual CPUs allocated to VM<br>• 4 GB RAM allocated to VM<br>• 1 Gbit Network interface card |
| Operating System | vSphere ESXi 5.5 as virtual machine host<br>VM image built with Windows Server 2012 R2 |

Explanation of Columns in Client Module Benchmark Results for explanation of the columns in the benchmark results tables.

**Table 11. One instance of Classification Running on a Single-Core Machine**

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| 5 templates (Automatic/HPA), Level 7 | 83,323 | 96% | 153 | 943 / 39 | 2 / 1135 |
| 500 templates (Automatic/HPA), Level 7 | 61,649 | 82% | 163 | 700 / 29 | 17 / 831 |
| 3000 templates (Automatic/HPA), Level 7 | 34,916 | 86% | 219 | 372 / 15 | 438 / 432 |
| 5 templates (Keyword Classification), Level 7, Western OCR (Full-text) | 1,017 | 99% | 403 | 42 / 15 | 25 / 39 |
| 5 templates (Keyword Classification), Level 7, East Euro / APAC OCR | 499 | 99% | 578 | 32 / 10 | 17 / 39 |
| 5 templates (Keyword Classification), Level 7, Advanced OCR/ICR | 1,555 | 99% | 488 | 45 / 7 | 27 / 63 |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| 5 templates (Keyword Classification), Level 7, General -Use OCR/ICR (Machine Print, Full-text, English, ACCURATE) | 583 | 99% | 489 | 34 / 10 | 4 / 34 |
| 100 templates (Text Matching), Level 7, Western OCR (Full-text) | 1,186 | 97% | 329 | 28 / 26 | 132 / 80 |
| 100 templates (Text Matching), Level 7, East Euro / APAC OCR | 547 | 100% | 370 | 16 / 18 | 35 / 59 |
| 100 templates (Text Matching), Level 7, Advanced OCR/ICR | 2,228 | 97% | 456 | 46 / 12 | 66 / 34 |
| 100 templates (Text Matching), Level 7, General-Use OCR/ICR (Machine Print, Full-text, English, ACCURATE) | 550 | 100% | 604 | 46 / 14 | 7 / 37 |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| 5 templates (Automatic/HPA), colored images, Level 7 | 13,858 | 96% | 165 | 2,762 / 20 | 11 / 2776 |

**Table 12. Multiple Instances of Classification Running on a Multi-Core Machine**

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second) (KB) |
|---|---|---|---|---|---|
| **5 templates (Automatic/HPA), Level 7** | | | | | |
| 1 Instance, 1 CPU core | 83,323 | 1.00 | 96% | 153 | 943 / 39 |
| 2 Instances, 2 CPU cores | 140,299 | 1.68 | 84% | 157 | 1,582 / 67 |
| 4 Instances, 4 CPU cores | 158,118 | 1.90 | 84% | 296 | 1,785 / 77 |
| **500 templates (Automatic/HPA), Level 7** | | | | | |
| 1 Instance, 1 CPU core | 61,649 | 1.00 | 82% | 163 | 700 / 29 |
| 2 Instances, 2 CPU cores | 113,905 | 1.84 | 86% | 170 | 1,261 / 54 |
| 4 Instances, 4 CPU cores | 144,651 | 2.35 | 86% | 336 | 1,586 / 71 |
| **3000 templates (Automatic/HPA), Level 7** | | | | | |
| 1 Instance, 1 CPU core | 34,916 | 1.00 | 89% | 219 | 372 / 15 |
| 2 Instances, 2 CPU cores | 62,515 | 1.79 | 89% | 272 | 819 / 35 |

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second) (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 88,696 | 2.54 | 82% | 580 | 1003 / 44 |
| **5 templates (Keyword Classification), Level 7, Western OCR (Full-text)** | | | | | |
| 1 Instance, 1 CPU core | 1,017 | 1.00 | 99% | 403 | 42 / 15 |
| 2 Instances, 2 CPU cores | 2,306 | 2.27 | 98% | 635 | 76 / 38 |
| 4 Instances, 4 CPU cores | 4,856 | 4.77 | 98% | 1296 | 68 / 69 |
| **5 templates (Keyword Classification), Level 7, East Euro / APAC OCR** | | | | | |
| 1 Instance, 1 CPU core | 499 | 1.00 | 99% | 578 | 32 / 10 |
| 2 Instances, 2 CPU cores | 1,127 | 2.26 | 99% | 963 | 18 / 28 |
| 4 Instances, 4 CPU cores | 1,583 | 3.17 | 67% | 2010 | 21 / 31 |
| **5 templates (Keyword Classification), Level 7, Advanced OCR/ICR** | | | | | |
| 1 Instance, 1 CPU core | 1,555 | 1.00 | 99% | 488 | 46 / 12 |
| 2 Instances, 2 CPU cores | 3,293 | 2.12 | 99% | 798 | 35 / 7 |
| 4 Instances, 4 CPU cores | 6,226 | 4.00 | 92% | 1584 | 76 / 26 |
| **5 templates, (Keyword Classification), Level 7, General-Use OCR/ICR (Machine Print, Full-text, English, ACCURATE)** | | | | | |
| 1 Instance, 1 CPU core | 583 | 1.00 | 99% | 489 | 34 / 10 |
| 2 Instances, 2 CPU cores | 1,224 | 2.10 | 99% | 728 | 17 / 20 |

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second) (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 2,525 | 4.33 | 98% | 1498 | 33 / 45 |
| **100 templates (Text Matching), Level 7, Western OCR (Full-text)** | | | | | |
| 1 Instance, 1 CPU core | 1,186 | 1.00 | 97% | 329 | 28 / 26 |
| 2 Instances, 2 CPU cores | 2,236 | 1.89 | 97% | 521 | 50 / 50 |
| 4 Instances, 4 CPU cores | 4,588 | 3.87 | 97% | 1108 | 95 / 105 |
| **100 templates (Text Matching), Level 7, East Euro / APAC OCR** | | | | | |
| 1 Instance, 1 CPU core | 547 | 1.00 | 100% | 370 | 16 / 18 |
| 2 Instances, 2 CPU cores | 1,056 | 1.93 | 99% | 898 | 27 / 35 |
| 4 Instances, 4 CPU cores | 1,360 | 2.49 | 66% | 1048 | 38 / 45 |
| **100 templates (Text Matching), Level 7, Advanced OCR/ICR** | | | | | |
| 1 Instance, 1 CPU core | 2,228 | 1.00 | 97% | 456 | 46 / 12 |
| 2 Instances, 2 CPU cores | 3,703 | 1.66 | 87% | 701 | 77 / 21 |
| 4 Instances, 4 CPU cores | 5,194 | 2.33 | 69% | 1512 | 106 / 30 |
| **100 templates (Text Matching), Level 7, General-Use OCR/ICR (Machine Print, Full-text, English, ACCURATE)** | | | | | |
| 1 Instance, 1 CPU core | 550 | 1.00 | 100% | 604 | 15 / 15 |
| 2 Instances, 2 CPU cores | 1,045 | 1.90 | 98% | 652 | 25 / 27 |

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second) (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 2.042 | 3.71 | 94% | 1308 | 43 / 51 |
| **5 templates (Automatic/HPA), Colored Images, Level 7** | | | | | |
| 1 Instance, 1 CPU core | 13,858 | 1.00 | 96% | 165 | 2,762 / 20 |
| 2 Instances, 2 CPU cores | 23,438 | 1.73 | 93% | 180 | 4,529 / 33 |
| 4 Instances, 4 CPU cores | 36,192 | 2.61 | 93% | 348 | 7,195 / 60 |

Refer to *Test Environment Used for Testing of Client Modules and Administrator* for details of the hardware and software used to gather these benchmark results and *Image Sets and Settings Used* for the image sets used for the Classification benchmark testing.

## Summary of Results

- Classification is CPU-intensive.
- Deploying additional instances of Classification on multi-core computers increases throughput by varying amounts. For example, for HPA level 7, with 2 instances running on 2 CPU cores, throughput is 168% of 1 instance. See section *Critical Factors Affecting Classification Performance and Tuning* for additional details.

## Critical Factors Affecting Classification Performance and Tuning

- **Classification type**: Classification performs the fastest with Automatic and HPA templates. Text matching and keyword templates are significantly slower.
- **Number of templates**: The number of Automatic and HPA templates significantly influences performance. The fewer the templates, the faster the processing. A project with 3000 Automatic or HPA templates processes the same images 1.77 times slower than a project with only 500 Automatic and HPA templates.
- **CPU speed**: All classification types are CPU-intensive. Fast CPUs are recommended.
- **CPU cores**: Multi-core machines are recommended, even though Classification is single-threaded.
- **Multiple instances on the same machine**: On a multi-core machine, deploy as many Classification instances as cores. On a single-core machine, deploy only one instance of Classification.
  - Because Advanced OCR licensing restricts throughput to 400 characters/second and 3600 documents/hour for all Classification instances running on a single computer, to achieve the desired scaling install only one instance of Classification per computer.
  - Because East Euro / APAC OCR licensing automatically assigns all Classification instances running on a computer to the same two CPU cores regardless of the number of available cores, to achieve the desired scaling manually assign each Classification instance to a different CPU core or deploy each Classification instance on a separate computer.
- **Network speed/Latency**: Local Area Network (LAN) is recommended rather than Wide Area Network (WAN).

## Sizing Recommendations

A single CPU machine and one instance of Classification can process 499 – 83,323 pages per hour, depending on the classification method and the project size (which are the critical factors). An additional instance of Classification deployed with an additional CPU increases throughput by 70% to 100%.

### To Estimate the Number of CPU Cores and Number of Classification Modules Required

- Estimate the number of pages per hour you will be processing.

- Determine the Classification methods which are used (graphical templates / OCR based classification) and select the benchmark result that most closely matches your usage.
- Determine the speed of the CPU you will be using in GHz and estimate its speed relative to the CPU used during the benchmark. Because Classification is CPU intensive, CPU speed significantly influences throughput.

## Sizing Formula

**Throughput** = (benchmark results for specific classification type * 75%) * (1 - ((2.5 - your CPU speed in GHz) / 2.5))

**Note:** The formula also uses only 75% of the benchmark result to normalize the benchmark results to a typical production environment.

## Example of Sizing Based on Environment

You have the following processing requirement and want to calculate the number of Classification modules you require to meet your processing needs:

- 5,000 pages per hour throughput requirement
- 50% of your pages require Automatic and HPA classification with 500 templates
- 30% of your pages require Keyword classification with 5 templates (Western OCR)
- 20% of your pages require Text Matching classification with 100 templates (Western OCR)
- Triggered at level 7
- CPU cores are 2.0 GHz

Using the formula, calculate the expected throughput of your CPU core and the number of CPU cores that are needed to run Classification three separate times, once for each classification type. Then add up the number of CPU cores to determine the number of Classification instances needed. Calculations will result in computing the portions of CPU processing capability required for each classification type. Note that CPU power can be shared among different types of classification tasks.

### Calculate Number of CPU Cores Required for Automatic and HPA Classification

- **Throughput (single instance)**: Throughput with 1 CPU core and 1 Classification instance: (61649 * 75%) * (1 - ((2.5 - 2.0) / 2.5)) = 36989
- **Number of CPU cores required**: (5000 * 50%) / 36989 = 0.07

### Calculate Number of CPU Cores Required for Keyword Classification

- **Throughput (single instance)**: Throughput with 1 CPU core and 1 Classification instance: (1017 * 75%) * (1 - ((2.5 - 2.0) / 2.5)) = 610
- **Number of CPU cores required**: (5000 * 30%) / 610 = 2.5

### Calculate Number of CPU Cores Required for Text Matching

- **Throughput (single instance)**: Throughput with 1 CPU core and 1 Classification instance: (1186 * 75%) * (1 - ((2.5 - 2.0) / 2.5)) = 712
- **Number of CPU cores required**: (5000 * 20%) / 712 = 1.4

### Calculate Total Number of CPU Cores Required

- **Total number of CPU Cores required:** 0.07 + 2.5 + 1.4 = 3.97 (round up to 4)
- **Total Number of Classification instances:** 4

## Extraction

Extraction performs automated data extraction from documents using optical character recognition (OCR).

### Test Scenarios

The following projects were configured with the HPA templates and benchmarked:

**Table 13. Projects Configured with the HPA Templates and Benchmarked**

| Test Scenario | Recognition Type | Field Type | Number of Fields | Engines |
|---|---|---|---|---|
| 1 | Zonal | Index | 5 | Western OCR |
| 2 | Zonal | Index | 10 | Western OCR |
| 3 | Zonal | Index | 10 | General-Use OCR/ICR |
| 4 | Zonal | Index | 10 | East Euro / APAC OCR |
| 5 | Zonal | Index | 10 | Advanced OCR/ICR |
| 6 | Zonal | Index<br>Table | Index: 10<br>Table: 5 | Western OCR |
| 7 | Free Form | Index | 5 | Western OCR |
| 8 | Free Form | Index | 10 | Western OCR |
| 9 | Free Form | Index | 10 | General-Use OCR/ICR |
| 10 | Free Form | Index | 10 | East Euro / APAC OCR |
| 11 | Free Form | Index | 10 | Advanced OCR/ICR |
| 12 | Free Form+LIFFE | Index<br>Table | Index: 10<br>Table: 5 | Western OCR |

### Benchmark Results

Refer to *Test Environment Used for Testing of Classification and* Extraction Modules

System Properties for Capture Client Modules and Administrator

| Item | Required |
|---|---|

| Item | Required |
|---|---|
| Hardware | Lenovo C30 Workstation<br>• Intel Xeon E5-2609 v2 @ 2.5 GHz (quad core)<br>• 4 virtual CPUs allocated to VM<br>• 4 GB RAM allocated to VM<br>• 1 Gbit Network interface card |
| Operating System | vSphere ESXi 5.5 as virtual machine host<br>VM image built with Windows Server 2012 R2 |

Explanation of Columns in Client Module Benchmark Results for explanation of the columns in the benchmark result tables.

**Table 14. One Instance of Extraction Running on a Single-Core Machine**

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| Zonal Recognition: 5 index zones (Western OCR) | 19,975 | 80% | 100 | 283 / 42 | 4 / 209 |
| Zonal Recognition: 10 index zones (Western OCR) | 13,806 | 88% | 103 | 201 / 43 | 3 / 151 |
| Zonal Recognition: 10 index zones (General-Use OCR/ICR) | 11,146 | 89% | 113 | 165 / 35 | 7 / 145 |
| Zonal Recognition: 10 index zones (East Euro / APAC OCR) | 7,362 | 94% | 171 | 115 / 24 | 13 / 117 |
| Zonal Recognition: 10 index zones (Advanced OCR/ICR) | 12,128 | 83% | 251 | 174 / 37 | 41 / 194 |
| Zonal Recognition: 10 index zones + 5 table fields (Western OCR) | 8,204 | 92% | 118 | 126 / 43 | 12 / 155 |
| Full Page Recognition: 5 index zones (Western OCR) | 1,159 | 100% | 205 | 31 / 18 | 3 / 59 |
| Full Page Recognition: 10 index zones (Western OCR) | 1,054 | 100% | 198 | 27 / 17 | 1 / 62 |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| Full Page Recognition: 10 index zones (General-Use OCR/ICR) | 796 | 99% | 199 | 19 / 13 | 41 / 78 |
| Full Page Recognition: 10 index zones (East Euro / APAC OCR) | 445 | 100% | 299 | 16 / 9 | 1 / 36 |
| Full Page Recognition: 10 index zones (Advanced OCR/ICR) | 650 | 83% | 1099 | 20 / 10 | 2 / 39 |
| Full Page Recognition: 10 index zones + 5 table fields LIFFE (Western OCR) | 913 | 98% | 224 | 40 / 19 | 3 / 64 |

**Table 15. Multiple Instances of Extraction Running on a Multi-Core Machine**

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| **Zonal Recognition: 5 index zones (Western OCR)** | | | | | |
| 1 Instance, 1 CPU core | 19,975 | 1.00 | 80% | 100 | 283 / 42 |
| 2 Instances, 2 CPU cores | 41,814 | 2.09 | 76% | 137 | 563 / 84 |

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 92,590 | 4.63 | 76% | 266 | 794 / 177 |
| **Zonal Recognition: 10 index zones (Western OCR)** | | | | | |
| 1 Instance, 1 CPU core | 13,806 | 1.00 | 88% | 103 | 201 / 43 |
| 2 Instances, 2 CPU cores | 26,434 | 1.91 | 83% | 142 | 374 / 82 |
| 4 Instances, 4 CPU cores | 56,833 | 4.12 | 83% | 274 | 793 / 177 |
| **Zonal Recognition: 10 index zones (General-Use OCR/ICR)** | | | | | |
| 1 Instance, 1 CPU core | 11,146 | 1.00 | 89% | 113 | 165 / 35 |
| 2 Instances, 2 CPU cores | 24,161 | 2.17 | 86% | 159 | 331 / 75 |
| 4 Instances, 4 CPU cores | 46,427 | 4.17 | 86% | 314 | 667 / 145 |
| **Zonal Recognition: 10 index zones (East Euro / APAC OCR)** | | | | | |
| 1 Instance, 1 CPU core | 7,362 | 1.00 | 94% | 171 | 115 / 24 |
| 2 Instances, 2 CPU cores | 16,638 | 2.26 | 89% | 272 | 230 / 52 |
| 4 Instances, 4 CPU cores | 23,881 | 3.24 | 70% | 538 | 359 / 75 |
| **Zonal Recognition: 10 index zones (Advanced OCR/ICR)** | | | | | |
| 1 Instance, 1 CPU core | 8,204 | 1.00 | 92% | 118 | 126 / 43 |
| 2 Instances, 2 CPU cores | 15,551 | 1.90 | 84% | 160 | 232 / 82 |

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 22,114 | 2.70 | 84% | 308 | 472/ 168 |
| **Zonal Recognition: 10 index zones + 5 table fields (Western OCR)** | | | | | |
| 1 Instance, 1 CPU core | 1,159 | 1.00 | 100% | 205 | 31 / 18 |
| 2 Instances, 2 CPU cores | 2,268 | 1.96 | 94% | 245 | 51 / 32 |
| 4 Instances, 4 CPU cores | 4,590 | 3.96 | 93% | 478 | 98 / 67 |
| **Full Page Recognition: 5 index zones (Western OCR)** | | | | | |
| 1 Instance, 1 CPU core | 1,159 | 1.00 | 100% | 205 | 31 / 18 |
| 2 Instances, 2 CPU cores | 2,268 | 1.96 | 94% | 245 | 51 / 32 |
| 4 Instances, 4 CPU cores | 4,590 | 3.96 | 93% | 478 | 98 / 67 |
| **Full Page Recognition: 10 index zones (Western OCR)** | | | | | |
| 1 Instance, 1 CPU core | 1,054 | 1.00 | 100% | 198 | 27 / 17 |
| 2 Instances, 2 CPU cores | 1,970 | 1.87 | 94% | 242 | 52 / 33 |
| 4 Instances, 4 CPU cores | 3,947 | 3.74 | 93% | 470 | 93 / 63 |
| **Full Page Recognition: 10 index zones (General-Use OCR/ICR)** | | | | | |
| 1 Instance, 1 CPU core | 796 | 1.00 | 99% | 199 | 19 / 13 |
| 2 Instances, 2 CPU cores | 1,475 | 1.85 | 98% | 246 | 31 / 25 |

| Instances | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 2,968 | 3.73 | 98% | 394 | 49 / 47 |
| **Full Page Recognition: 10 index zones (East Euro / APAC OCR)** | | | | | |
| 1 Instance, 1 CPU core | 445 | 1.00 | 100% | 299 | 16 / 9 |
| 2 Instances, 2 CPU cores | 873 | 1.96 | 99% | 444 | 33 / 20 |
| 4 Instances, 4 CPU cores | 1,258 | 2.83 | 70% | 866 | 53 / 26 |
| **Full Page Recognition: 10 index zones (Advanced OCR/ICR)** | | | | | |
| 1 Instance, 1 CPU core | 650 | 1.00 | 83% | 1099 | 20 / 10 |
| 2 Instances, 2 CPU cores | 1,059 | 1.63 | 58% | 2246 | 36 / 20 |
| 4 Instances, 4 CPU cores | 1,200 | 1.85 | 32% | 4434 | 52 / 22 |
| **Full Page Recognition: 10 index zones + 5 table fields LIFFE (Western OCR)** | | | | | |
| 1 Instance, 1 CPU core | 913 | 1.00 | 98% | 224 | 40 / 19 |
| 2 Instances, 2 CPU cores | 1,668 | 1.83 | 94% | 267 | 44 / 35 |
| 4 Instances, 4 CPU cores | 3,387 | 3.71 | 92% | 526 | 82 / 62 |

## Summary of Results

- Running one instance per CPU core typically improves throughput by 90%.
- In some cases, running multiple instances of the module does not improve the throughput due to the OCR engine licensing restriction. For example, see the test cases where Advanced OCR/ICR and East Euro / APAC OCR were used.
- The number of fields that need to be extracted impacts the module throughput rate. The fewer fields, the faster is the rate of extraction. This applies to both Zonal and Free Form Extraction test cases.

## Critical Factors Affecting Extraction Performance

- **Zonal vs. Full Page recognition:** Captiva Extraction performs faster using anchored zones compared to Free Form and LIFFE.
- **OCR engine:** Captiva Extraction shows better throughput when using Western OCR to extract fields and table fields data. Throughput decreases significantly when using General-Use OCR/ICR, East Euro / APAC OCR, and Advanced OCR/ICR.
- **Number of extracted fields:** The number of fields that need to be extracted significantly influences performance. The fewer the fields, the faster the processing. This applies to both Zonal and Free Form extraction scenarios.
- **CPU cores:** Captiva Extraction is CPU-sensitive. Multi-core machines are recommended, even though Extraction is single-threaded. Do not run more instances of Captiva Extraction than the number of available CPUs in the system.
- **CPU speed:** All classification types are CPU-intensive. Fast CPUs are recommended.
  - **Advanced OCR/ICR**: When the Advanced OCR/ICR engine is used, Advanced OCR/ICR licensing restricts throughput for all instances of Extraction running on the machine. To achieve desired scaling it may be necessary to deploy additional instances of Extraction on separate machines.
  - **East Euro / APAC OCR**: When the East Euro / APAC OCR engine is used, East Euro / APAC OCR licensing restrict all instances of Extraction running on the machine to the same two CPU cores regardless of the number of available cores. To achieve desired scaling, deploy additional instances of the Extraction on separate machines.
- **Machine memory:** Captiva Extraction consumes up to 450 MB of RAM when extracting field data using Full Page recognition, and about 100 MB of RAM when using Zonal recognition. A minimum of 500 MB RAM for each instance of Captiva Extraction is recommended.
- **Network speed/Latency**: Local Area Network (LAN) is recommended rather than Wide Area Network (WAN).

## Non-Critical Factors

- **Disk usage**: Disk usage is a minor factor since the module deletes disk space used after finishing a task. 50 GB free disk space is recommended.

## Sizing Recommendations

A single-core machine with one instance of Captiva Extraction can typically process 445 – 19,975 pages per hour, depending on the OCR engine used to extract field data. Each additional instance of Extraction deployed on a dual-core machine increases throughput 63% to 126%.

# Image Converter

Image Converter module performs image file conversion to a different file format, image properties modification, file merging and splitting, and "burning" of annotations added by other modules such as Image Processor or Captiva Completion.

## Test Scenarios

The following scenarios were performed to test various document sizes, image sizes, and file types.

**Table 16. Image Converter Splitting Scenarios**

| Scenario | Description |
|---|---|
| **Scenario 1:**<br>Input = multipage image (binary TIFF, 300 DPI)<br>Output = 10 pages (binary TIFF, 300 DPI) | Files processed are 300 DPI, binary, multipage TIFF files (10 pages each), stored at the document level. These files are split into 10 page-level images which are also TIFF, binary, 300 DPI. |
| **Scenario 2:**<br>Input = multipage image (binary TIFF, 300 DPI)<br>Output = 100 pages (binary TIFF, 300 DPI) | Files processed are 300 DPI, binary, multipage TIFF files (100 pages each), stored at the document level. These files are split into 100 page-level images which are also TIFF, binary, 300 DPI. |
| **Scenario 3:**<br>Input = multipage image (color TIFF, 300 DPI)<br>Output = 10 pages (color TIFF, 300 DPI) | Files processed are 300 DPI, 24-bit color, multipage TIFF files (10 pages each), stored at the document level. These files are split into 10 page-level images which are also TIFF, 24-bit color, 300 DPI. |
| **Scenario 4:**<br>Input = multipage image (color TIFF, 300 DPI)<br>Output = 100 pages (color TIFF, 300 DPI) | Files processed are 300 DPI, 24-bit, color multipage TIFF files (100 pages each), stored at the document level. These files are split into 100 page-level images which are also TIFF, 24-bit color, 300 DPI. |
| **Scenario 5:**<br>Input = multipage color PDF (non-image)<br>Output = 10 pages (color TIFF, 300 DPI) | Files processed are color, non-image PDF files (10 pages each), stored at the document level. These files are rendered and converted into 10 page-level images which are TIFF, 24-bit color, 300 DPI. |

| Scenario | Description |
|---|---|
| **Scenario 6:**<br>Input = multipage color PDF (non-image)<br>Output = 100 pages (color TIFF, 300 DPI) | Files processed are color, non-image PDF files (100 pages each), stored at the document level. These files are rendered and converted into 100 page-level images which are TIFF, 24-bit color, 300 DPI. |
| **Scenario 7:**<br>Input = multipage Word DOCX file (10 pages)<br>Output = 10 pages (binary TIFF, 300 DPI) | Files processed are Word documents in DOCX format (10 pages each) and stored at document level. These files are rendered in Word 2013 and converted into 10 page-level images which are TIFF, binary, 300 DPI. |
| **Scenario 8:**<br>Input = multipage Word DOCX file (100 pages)<br>Output = 100 pages (binary TIFF, 300 DPI) | Files processed are Word documents in DOCX format (100 pages each) and stored at document level. These files are rendered in Word 2013 and converted into 100 page-level images which are TIFF, binary, 300 DPI. |
| **Scenario 9:**<br>Input = multipage Excel XLSX file (10 pages)<br>Output = 10 pages (binary TIFF, 300 DPI) | Files processed are Excel files in XLSX format (10 pages each) and stored at document level. These files are rendered in Excel 2013 and converted into 10 page-level images which are TIFF, binary, 300 DPI. |
| **Scenario 10:**<br>Input = multipage Excel XLSX file (100 pages)<br>Output = 100 pages (binary TIFF, 300 DPI) | Files processed are Excel files in XLSX format (100 pages each) and stored at document level. These files are rendered in Excel 2013 and converted into 100 page-level images which are TIFF, binary, 300 DPI. |

**Table 17. Image Converter Consolidating Scenarios**

| Scenario | Description |
|---|---|
| **Scenario 11:**<br>Input = 10 single page images (binary TIFF, 300 DPI)<br>Output = 1 multipage image (binary TIFF, 300 DPI) | Consolidates 10 page-level 300 DPI binary TIFF images into one document-level multipage TIFF, also 300 DPI binary. |

| Scenario | Description |
|---|---|
| **Scenario 12:**<br>Input = 100 single page images (binary TIFF, 300 DPI)<br>Output = 1 multipage image (binary TIFF, 300 DPI) | Consolidates 100 page-level 300 DPI binary TIFF images into one document-level multipage TIFF, also 300 DPI binary. |
| **Scenario 13:**<br>Input = 10 single page images (color TIFF, 300 DPI)<br>Output = 1 multipage image (color TIFF, 300 DPI) | Consolidates 10 page-level 300 DPI color (24-bit) TIFF images into one document-level multipage TIFF, also 300 DPI color. |
| **Scenario 14:**<br>Input = 100 single page images (color TIFF, 300 DPI)<br>Output = 1 multipage image (color TIFF, 300 DPI) | Consolidates 100 page-level 300 DPI color (24-bit) TIFF images into one document-level multipage TIFF, also 300 DPI color. |
| **Scenario 15:**<br>Input = 10 single page PDF files (color)<br>Output = 1 multipage PDF file (color) | Consolidates 10 page-level color PDF “image + hidden text” files into one document-level multipage PDF file. |
| Scenario 16:<br>Input = 100 single page PDF files (color)<br>Output = 1 multipage PDF file (color) | Consolidates 100 page-level color PDF “image + hidden text” files into one document-level multipage PDF file. |
| **Scenario 17:**<br>Input = 10 single page PDF files (binary)<br>Output = 1 multipage PDF file (binary) | Consolidates 10 page-level binary PDF “image + hidden text” files into one document-level multipage PDF file. |
| **Scenario 18:**<br>Input = 100 single page PDF files binary)<br>Output = 1 multipage PDF file (binary) | Consolidates 100 page-level binary PDF “image + hidden text” files into one document-level multipage PDF file. |

| Scenario | Description |
|---|---|
| **Scenario 19:**<br>Input = 1000 single age PDF files (binary)<br>Output = 1 multipage PDF file (binary) | Consolidates 1000 page-level binary PDF "image + hidden text" files into one document-level multipage PDF file. |

## Benchmark Results

The scenarios described previously were performed to test various document sizes, image sizes, and file types:

- Splitting scenarios to test splitting multi-page files into individual single-page TIFF/G4 images.
- Consolidating scenarios to test merging single page images to a multi-page document.

**Table 18. One instance of Image Converter Running on a Single-Core Machine**

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| **Splitting Scenarios** | | | | | |
| Binary TIFF/G4, 10 pages/doc, 100 docs/batch, 75 KB average image size | 135,211 | 62 | 46 | 3,004 | 3,227 |
| Binary TIFF/G4, 100 pages/doc, 10 docs/batch, 75 KB average image size | 164,103 | 83 | 73 | 3,675 | 4,050 |
| Color TIFF /G4, 10 pages/doc, 100 docs/batch, 1 MB average image size | 9,367 | 98 | 94 | 3,675 | 4,050 |
| Color TIFF /G4, 100 pages/doc, 10 docs/batch, 1 MB average image size | 9,318 | 100 | 95 | 2,781 | 2,884 |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Color non - image PDF, 10 pages/doc, 100 docs /batch, N/A average image size | 5,125 | 100 | 98 | 1,569 | 1,914 |
| Color non - image PDF, 100 pages/doc, 10 docs/batch, N/A average image size | 6,076 | 100 | 129 | 122 | 987 |
| Color DOCX, 10 pages/doc, 100 docs /batch, N/A average image size | 3,288 | 100 | 76 | 249 | 667 |
| Color DOCX, 100 pages/doc, 10 docs/batch, N/A average image size | 3,897 | 100 | 85 | 81 | 643 |
| XLSX, 10 ages/doc, 100 docs/batch, N/A average image size | 3,288 | 100 | 76 | 238 | 631 |
| XLSX, 100 ages/doc, 10 docs/batch, N/A average image size | 4,782 | 100 | 96 | 29 | 1,154 |
| **Consolidating Scenarios** | | | | | |
| Binary TIFF/G4, 10 pages/doc, 100 docs/batch, 75 KB average image size | 203,851 | 57 | 54 | 4,425 | 4,342 |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Binary TIFF/G4, 100 pages/doc, 10 docs/batch, 75 KB average image size | 163,636 | 70 | 67 | 3,731 | 3,669 |
| Color TIFF /G4, 10 pages/doc, 100 docs/batch, 1 MB average image size | 23,057 | 87 | 80 | 6,715 | 6,619 |
| Color TIFF /G4, 100 pages/doc, 10 docs/batch, 1 MB average image size | 23,097 | 86 | 79 | 7,043 | 6,808 |
| Color PDF, 10 pages/doc, 100 docs/batch, 1 MB average image size | 31,395 | 62 | 50 | 18,926 | 18,850 |
| Color PDF, 100 pages/doc, 10 docs/batch, 1 MB average image size | 26,846 | 59 | 47 | 16,493 | 15,724 |
| Binary PDF, 10 pages/doc, 100 docs/batch, 75 KB average image size | 117,647 | 74 | 71 | 3,393 | 2,509 |
| Binary PDF, 100 pages/doc, 10 docs/batch, 75 KB average image size | 122,449 | 72 | 72 | 3,091 | 2,945 |
| Binary PDF, 1000 pages/doc, 1 docs/batch, 75 KB average image size | 36,400 | 96 | 91 | 982 | 930 |

Table 19. Two Instances of Image Converter Running on a Dual-Core Machine

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| **Splitting Scenarios** | | | | | |
| Binary TIFF/G4, 10 pages/doc, 100 docs/batch, 75 KB average image size | 244,068 | 60 | 60 | 5,442 | 5,877 |
| Binary TIFF/G4, 100 pages/doc, 10 docs/batch, 75 KB average image size | 306,383 | 80 | 73 | 6,859 | 7,554 |
| Color TIFF /G4, 10 pages/doc, 100 docs/batch, 1 MB average image size | 17,658 | 97 | 111 | 5,282 | 5,447 |
| Color TIFF /G4, 100 pages/doc, 10 docs/batch, 1 MB average image size | 16,112 | 99 | 306 | 4,786 | 4,913 |
| Color non - image PDF, 10 pages/doc, 100 docs /batch, N/A average image size | 10,159 | 98 | 127 | 562 | 1,908 |
| Color non - image PDF, 100 pages/doc, 10 docs/batch, N/A average image size | 11,734 | 100 | 115 | 235 | 1,911 |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Color DOCX, 10 pages/doc, 100 docs /batch, N/A average image size | 4,464 | 100 | 122 | 359 | 839 |
| Color DOCX, 100 pages/doc, 10 docs/batch, N/A average image size | 4,745 | 74 | 111 | 88 | 771 |
| XLSX, 10 pages/doc, 100 docs/batch, N/A average image size | 3,655 | 63 | 114 | 340 | 840 |
| XLSX, 100 pages/doc, 10 docs/batch, N/A average image size | 5,065 | 72 | 106 | 120 | 1,321 |
| **Consolidating Scenarios** | | | | | |
| Binary TIFF/G4, 10 pages/doc, 100 docs/batch, 75 KB average image size | 342,857 | 55 | 98 | 4,946 | 5,491 |
| Binary TIFF/G4, 100 pages/doc, 10 docs/batch, 75 KB average image size | 409,091 | 50 | 76 | 9,336 | 9,167 |
| Color TIFF /G4, 10 pages/doc, 100 docs/batch, 1 MB average image size | 46,693 | 80 | 99 | 12,688 | 13,528 |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Color TIFF /G4, 100 pages/doc, 10 docs/batch, 1 MB average image size | 47,556 | 82 | 108 | 14,239 | 13,834 |
| Color PDF, 10 pages/doc, 100 docs/batch, 1 MB average image size | 49,793 | 57 | 133 | 31,547 | 31,369 |
| Color PDF, 100 pages/doc, 10 docs/batch, 1 MB average image size | 46,693 | 55 | 262 | 29,800 | 29,396 |
| Binary PDF, 10 pages/doc, 100 docs/batch, 75 KB average image size | 165,517 | 59 | 172 | 8,261 | 3,478 |
| Binary PDF, 100 pages/doc, 10 docs/batch, 75 KB average image size | 215,569 | 57 | 147 | 5,420 | 5,140 |
| Binary PDF, 1000 pages/doc, 1 docs/batch, 75 KB average image size | 53,812 | 95 | 268 | 1,535 | 1,356 |

**Table 20. Four Instances of Image Converter Running on a Quad-Core Machine**

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Splitting Scenarios | | | | | |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Binary TIFF/G4, 10 pages/doc, 100 docs/batch, 75 KB average image size | 378,947 | 76 | 269 | 11,432 | 12,433 |
| Binary TIFF/G4, 100 pages/doc, 10 docs/batch, 75 KB average image size | 507,042 | 77 | 76 | 11,431 | 12,551 |
| Color TIFF /G4, 10 pages/doc, 100 docs/batch, 1 MB average image size | 32,877 | 98 | 100 | 9,848 | 10,211 |
| Color TIFF /G4, 100 pages/doc, 10 docs/batch, 1 MB average image size | 28,162 | 100 | 290 | 8,346 | 8,559 |
| Color non - image PDF, 10 pages/doc, 100 docs /batch, N/A average image size | 18,677 | 99 | 116 | 980 | 3,583 |
| Color non - image PDF, 100 pages/doc, 10 docs/batch, N/A average image size | 21,570 | 100 | 125 | 439 | 3,525 |
| Color DOCX, 10 pages/doc, 100 docs /batch, N/A average image size | 4,422 | 37 | 107 | 358 | 834 |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Color DOCX, 100 pages/doc, 10 docs/batch, N/A average image size | 4,708 | 32 | 115 | 91 | 760 |
| XLSX, 10 pages/doc, 100 docs/batch, N/A average image size | 4,235 | 37 | 105 | 352 | 821 |
| XLSX, 100 pages/doc, 10 docs/batch, N/A average image size | 4,968 | 28 | 99 | 124 | 1,306 |
| **Consolidating Scenarios** | | | | | |
| Binary TIFF/G4, 10 pages/doc, 100 docs/batch, 75 KB average image size | 437,630 | 39 | 73 | 9,566 | 9,479 |
| Binary TIFF/G4, 100 pages/doc, 10 docs/batch, 75 KB average image size | 397,146 | 33 | 75 | 9,026 | 8,867 |
| Color TIFF /G4, 10 pages/doc, 100 docs/batch, 1 MB average image size | 62,348 | 61 | 97 | 18,575 | 18,394 |
| Color TIFF /G4, 100 pages/doc, 10 docs/batch, 1 MB average image size | 65,544 | 64 | 109 | 19,687 | 19,344 |

| Scenarios | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received per Second (KB) | Network Data Sent per |
|---|---|---|---|---|---|
| Color PDF, 10 pages/doc, 100 docs/batch, 1 MB average image size | 61,565 | 42 | 128 | 38,858 | 38,610 |
| Color PDF, 100 pages/doc, 10 docs/batch, 1 MB average image size | 60,632 | 43 | 251 | 39,014 | 38,300 |
| Binary PDF, 10 pages/doc, 100 docs/batch, 75 KB average image size | 350,195 | 60 | 111 | 10,835 | 89,037 |
| Binary PDF, 100 pages/doc, 10 docs/batch, 75 KB average image size | 388,769 | 60 | 131 | 9,714 | 9,083 |
| Binary PDF, 1000 pages/doc, 1 docs/batch, 75 KB average image size | 71,914 | 97 | 267 | 1,790 | 1,762 |

## Summary of Results

- With two instances of the module running on two CPU cores, throughput of the module is 171% of one instance running on one CPU core. So, running two copies of the module will improve the module throughput as long as the system has 2 or more CPU cores.
- Image Converter is limited to processing and creating a maximum stage file size of 250 MB.
- Tasks converting Microsoft Office files will only see a 15% gain for the second instance and none for subsequent instances of the module on the same machine due to Microsoft Office limitations.
- Image Converter can consume a large amount of memory if working with large stage files. 500 MB per instance is recommended.

## Critical Factors Affecting Image Converter Performance

- Image Converter can cause very high network traffic when processing color images, up to 39 MB/sec.
- Image Converter performance slows down as processed stage file increases.

- Image Converter uses more memory as the stage file size increases.

### Non-Critical Factors

- **Disk usage**: Disk usage is a minor factor since the module frees the disk space used after finishing a task. 50 GB of free disk space is recommended.

### Sizing Recommendations

A single-core machine and one instance of Image Converter can typically process 3,288 - 164,103 pages per hour, depending on the resolution, DPI, and file type. An additional instance of Image Converter deployed on a dual-core machine increases throughput by 6 % to 98%.

## Image Processor

Image Processor is used to improve image quality such as deskew, detect image features such as blank pages and patch codes, and to annotate TIFF images.

Typically, Image Processor is used prior to an OCR step to improve the OCR recognition accuracy.

### Test Scenarios

The following scenarios were performed to measure effect of various filters and filter sets applied to TIFF files with different characteristics, such as colored and bitonal images, files size, image size, compression, and DPI.

**Table 21. Image Processor Test Scenarios**

| Scenario | Description |
|---|---|
| **Scenario 1:**<br>Color and bitonal scripted cleanup (color only) | The goal consists in measuring effect of a reasonable mix of common filters, using color images only.<br>• Image set: *ColorImg_101.tif*<br>(Color TIFF image, 780 KB, using "old style" JPEG compression, 300 DPI, US letter sized color invoice "KABA ILCO" with a Code39 barcode)<br>• Profile: Color and Bitonal Scripted Cleanup<br>(sample, but all images treated as color)<br>• Filters: Convert Specific Color 1, Convert Specific Color 2, Convert Specific Color 3, Convert to Black-White, Smooth Edges, Remove Black Bars, Remove Lines, Remove Background, Remove Specks, Detect Barcodes, Detect Blank Pages |
| **Scenario 2:**<br>Color and bitonal scripted cleanup (bitonal only) | The goal consists in measuring effect of a reasonable mix of common filters, using binary images only.<br>• Image set: *Pageinvoicebin101.tif*<br>(Binary TIFF image, 52 KB, G4 compression, 300 DPI, US letter sized invoice ("Doodads") with a Code25_Interleaved and a PDF417 barcode)<br>• Profile: Color and Bitonal Scripted Cleanup<br>(sample, but all images treated as bitonal)<br>• Filters: Smooth Edges, Remove Black Bars, Remove Lines, Remove Background, Remove Specks, Detect Barcodes, Detect Blank Pages |

| Scenario | Description |
| --- | --- |
| **Scenario 3:**<br>Annotations only | The goal was measuring effect of doing only annotations (for instance, replacement for IAStamp).<br>• Image set: *Pageinvoicebin101.tif*<br>(Binary TIFF image, 52 KB, G4 compression, 300 DPI, US letter sized invoice ("Doodads") with a Code25_Interleaved and a PDF417 barcode)<br>• Profile: defined one of each type of annotation, with the text annotation inserting an IA value<br>• Filters: no |
| **Scenario 4:**<br>Deskew only | The goal was measuring the "best case" configuration.<br>• Image set: *Pageinvoicebin101.tif*<br>(Binary TIFF image, 52 KB, G4 compression, 300 DPI, US letter sized invoice ("Doodads") with a Code25_Interleaved and a PDF417 barcode)<br>• Profile: used the Deskew filter only.<br>• Filters: Deskew<br>**Note:** The images used did not contain any skew, so they were not really altered. |
| **Scenario 5:**<br>Scaling (color) | The goal consisted in measuring effect of DPI scaling.<br>• Image set: *300Mixedsizecol101.tif*<br>(set of color images at 300 DPI of various page sizes, such as US Letter size and smaller sizes)<br>• Profile: used the Scale filter only that was configured to Change<br>Resolution to 200 DPI (input images were 300 DPI)<br>• Filters: Scale |
| **Scenario 6:**<br>Scaling (binary) | The goal consisted in measuring effect of DPI scaling.<br>• Image set: *300Mixsizebat101.tif*<br>(set of binary images at 300 DPI of various page sizes)<br>• Profile: used the Scale filter only that was configured to Change<br>Resolution to 200 DPI (input images were 300 DPI)<br>• Filters: Scale |

## Benchmark Results

The following conditions apply to all test scenarios:

- Every instance of Image Processor was run on a separate CPU. Tests performed on “1 module/1 CPU”, “2 modules/2 CPU”, and “4 modules/4 CPU” configurations.
- The results of the “3 modules/3 CPU” configuration are interpolated as a arithmetic average of the “2 modules/2 CPU” and “4 modules/4 CPU” configuration values.
- The number of batches processed are 100 for “1 module/1 CPU” and “2 modules/2 CPU” configurations and 400 for the “4 modules/4 CPU” configuration.
- The number of pages processed are 10,000 for “1 module/1 CPU” and “2 modules/2 CPU” configurations and 40,000 for the “4 modules/4 CPU” configuration.

Table 22. **Image Processor Benchmark Results**

| Number of module instances running on separate CPU | Duration (minutes) | Throughput (pages/hr) | Effective Units of Single Modules |
|---|---|---|---|
| **Scenario 1:** Color and Bitonal Scripted Cleanup (color only). Trigger level : 0 | | | |
| 1 CPU, 1 module | 85.82 | 6,992 | 1.00 |
| 2 CPU, 2 modules | 48.57 | 12,354 | 1.77 |
| 3 CPU, 3 modules (interpolated) | N/A | 17,644 | 2.52 |
| 4 CPU, 4 modules | 104.65 | 22,934 | 3.28 |
| **Scenario 2:** Color and Bitonal Scripted Cleanup (bitonal only). Trigger level : 0 | | | |
| 1 CPU, 1 module | 93.75 | 6,400 | 1.00 |
| 2 CPU, 2 modules | 43.58 | 13,767 | 2.15 |
| 3 CPU, 3 modules (interpolated) | N/A | 20,001 | 3.13 |
| 4 CPU, 4 modules | 91.48 | 26234 | 4.10 |
| **Scenario 3:** Annotations only. Trigger level : 0 | | | |
| 1 CPU, 1 module | 45.40 | 13,216 | 1.00 |
| 2 CPU, 2 modules | 23.93 | 25,070 | 1.90 |
| 3 CPU, 3 modules (interpolated) | N/A | 32,261 | 2.44 |
| 4 CPU, 4 modules | 60.83 | 39,452 | 2.99 |
| **Scenario 4:** Deskew only. Trigger level : 0 | | | |
| 1 CPU, 1 module | 14.72 | 40,770 | 1.00 |
| 2 CPU, 2 modules | 7.77 | 77,253 | 1.89 |
| 3 CPU, 3 modules (interpolated) | N/A | 109,984 | 2.70 |
| 4 CPU, 4 modules | 16.82 | 142,716 | 3.50 |
| **Scenario 5:** Scaling (color). Trigger level : 0 | | | |
| 1 CPU, 1 module | 80.37 | 7,466 | 1.00 |
| 2 CPU, 2 modules | 40.45 | 14,833 | 1.99 |
| 3 CPU, 3 modules (interpolated) | N/A | 21,590 | 2.89 |
| 4 CPU, 4 modules | 84.67 | 28,346 | 3.80 |
| **Scenario 6:** Scaling (bitonal). Trigger level : 0 | | | |
| 1 CPU, 1 module | 28.53 | 21,028 | 1.00 |
| 2 CPU, 2 modules | 16.40 | 36,585 | 1.74 |
| 3 CPU, 3 modules (interpolated) | N/A | 50,696 | 2.41 |
| 4 CPU, 4 modules | 37.03 | 64,806 | 3.08 |

## Critical Factors Affecting Image Processor Performance

- **CPU intensive**: Image Processor is consuming nearly 99% of a CPU.
- **Filters**: Processing rate is highly dependent on which filters are used, how they are configured, and what sorts of images are used. A general conservative estimate of a fairly complex usage is about 7,000 pages per hour on a 3.0 GHz CPU, although in some very simple scenarios it can achieve up to 40,000 pages per hour.
- **Color images and high resolution images**: Expect color images and images with higher resolutions (more than 300 DPI) to process more slowly.
- **Multiple instances on the same machine:** On a multi-core machine, you can deploy additional instances of the module, provided there is at least one CPU available per module instance.

With multiple instances running on same system, each instance's throughput will be reduced by about 20%. For example, if 1 instance on one CPU achieves 10,000 pages per hour, expect 4 instances on four CPUs to achieve about 10,000 * 4 * 0.8 = about 32,000 pages per hour.

## Non-Critical Factors Affecting Image Processor Performance

- **Memory usage**: Observations of the "Private Bytes" during heavy module use with 300 DPI color images showed about 132 MB private memory used (per module instance). To be safe, add an additional 256 MB RAM for each instance that will run on a multi-core machine.
- **Disk usage**: Disk usage is a minor factor, as the module rarely uses the disk at all.

# NuanceOCR

NuanceOCR performs optical character recognition of scanned or imported images and exports the image and index data to different word processing and text formats.

## Test Scenarios

Full-page with PDF – ACCURATE (Trigger level 0)

Full-page with PDF – BALANCED (Trigger level 0)

Full-page with PDF – FAST (Trigger level 0)

Full-page with PDF – ACCURATE (Trigger level 1)

Scenarios 1, 2, 3 and 4 test the creation of a PDF file containing the scanned image and searchable, full page OCR results in the PDF file. This is the most CPU-intensive usage of NuanceOCR. This test was run with 3 different "Accuracy" settings to see how the accuracy setting affects throughput.

## Benchmark Results

Refer to *Test Environment Used for Testing of Classification and* Extraction Modules

System Properties for Capture Client Modules and Administrator

| Item | Required |
| --- | --- |

| Item | Required |
|---|---|
| Hardware | Lenovo C30 Workstation<br>• Intel Xeon E5-2609 v2 @ 2.5 GHz (quad core)<br>• 4 virtual CPUs allocated to VM<br>• 4 GB RAM allocated to VM<br>• 1 Gbit Network interface card |
| Operating System | vSphere ESXi 5.5 as virtual machine host<br>VM image built with Windows Server 2012 R2 |

Explanation of Columns in Client Module Benchmark Results for explanation of the columns in the benchmark results tables.

**Table 23. One Instance of NuanceOCR Running on a Single-Core Machine**

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| **Image Resolution 200 DPI** | | | | | |
| Full-page with PDF – ACCURATE (Trigger level 0) | 4,567 | 96% | 91 | 90 / 64 | 8 / 95 |
| Full-page with PDF – BALANCED (Trigger level 0) | 8,959 | 99% | 99 | 145 / 85 | 3 / 90 |
| Full-page with PDF – FAST (Trigger level 0) | 9,041 | 99% | 105 | 150 / 91 | 1 / 70 |
| Full-page with PDF – ACCURATE (Trigger level 1) | 5,673 | 99% | 113 | 37 / 36 | 1 / 61 |
| **Image Resolution 300 DPI** | | | | | |
| Full-page with PDF – ACCURATE (Trigger level 0) | 3,360 | 97% | 192 | 74 / 50 | 2 / 34 |
| Full-page with PDF – BALANCED (Trigger level 0) | 5,428 | 99% | 174 | 100 / 57 | 2 / 68 |
| Full-page with PDF – FAST (Trigger level 0) | 6,895 | 99% | 181 | 127 / 87 | 1 / 65 |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| Full-page with PDF – ACCURATE (Trigger level 1) | 3,795 | 99% | 198 | 32 / 29 | 1 / 43 |

**Table 24. Multiple Instances of NuaneOCR Running on a Multi-Core Machine**

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| **Image Resolution 200 DPI** | | | | | |
| Full-page with PDF – ACCURATE (Trigger level 0) | | | | | |
| 1 Instance, 1 CPU core | 4,567 | 1.00 | 96% | 91 | 90 / 64 |
| 2 Instances, 2 CPU cores | 9,224 | 2.01 | 95% | 182 | 185 / 132 |
| 4 Instances, 4 CPU cores | 17,543 | 3.84 | 94% | 374 | 185 / 132. |
| Full-page with PDF – BALANCED (Trigger level 0) | | | | | |
| 1 Instance, 1 CPU core | 8,959 | 1.00 | 99% | 99 | 145 / 85 |
| 2 Instances, 2 CPU cores | 17,402 | 1.94 | 96% | 182 | 286 / 170 |
| 4 Instances, 4 CPU cores | 32,547 | 3.63 | 98% | 364 | 286 / 170 |
| Full-page with PDF – FAST (Trigger level 0) | | | | | |
| 1 Instance, 1 CPU core | 9,041 | 1.00 | 99% | 105 | 150 / 91 |
| 2 Instances, 2 CPU cores | 17,585 | 1.94 | 97% | 182 | 289 / 176 |
| 4 Instances, 4 CPU cores | 34,028 | 3.76 | 99% | 364 | 289 / 176 |
| Full-page with PDF – ACCURATE (Trigger level 1) | | | | | |
| 1 Instance, 1 CPU core | 5,673 | 1.00 | 99% | 162 | 88 / 7 |
| 2 Instances, 2 CPU cores | 11,097 | 1.95 | 88% | 807 | 156 / 13 |

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 35,273 | 6.21 | 88% | 1664 | 419 / 34 |
| **Image Resolution 300 DPI** | | | | | |
| Full-page with PDF – ACCURATE (Trigger level 0) | | | | | |
| 1 Instance, 1 CPU core | 3,360 | 1.00 | 97 | 193 | 74 / 50 |
| 2 Instances, 2 CPU cores | 6,606 | 1.96 | 96% | 193 | 143 / 97. |
| 4 Instances, 4 CPU cores | 13,442 | 4.0 | 97% | 386 | 143 / 97 |
| Full-page with PDF – BALANCED (Trigger level 0) | | | | | |
| 1 Instance, 1 CPU core | 5,428 | 1.00 | 99% | 174 | 100 / 57 |
| 2 Instances, 2 CPU cores | 10,767 | 1.98 | 98% | 205 | 198 / 114 |
| 4 Instances, 4 CPU cores | 20,412 | 3.76 | 99% | 409 | 198 / 114 |
| Full-page with PDF – FAST (Trigger level 0) | | | | | |
| 1 Instance, 1 CPU core | 6,895 | 1.00 | 99% | 181 | 127 / 87 |
| 2 Instances, 2 CPU cores | 14,114 | 2.04 | 99% | 218 | 245 / 170 |
| 4 Instances, 4 CPU cores | 27,119 | 3.93 | 99% | 437 | 255 / 181 |
| Full-page with PDF – ACCURATE (Trigger level 1) | | | | | |
| 1 Instance, 1 CPU core | 3,795 | 1.00 | 99% | 197 | 32 / 29 |
| 2 Instances, 2 CPU cores | 8,429 | 2.22 | 99% | 237 | 123 / 95 |
| 4 Instances, 4 CPU cores | 14,267 | 3.75 | 99% | 474 | 181 / 95 |

## Summary of Results

- The module throughput can be impacted by the OCR engine accuracy setting.
- Triggering the module at level 1 yields higher throughput than triggered it at level 0.

## Critical Factors Affecting NuanceOCR Performance

- **The Recognition Engine settings**: Balanced and Fast settings yield higher throughput than Accurate. Throughput is 45% greater when the Recognition Engine setting is Balanced instead of Accurate (5,428 pages/hr instead of 3,360 pages/hr). However, the recognition accuracy is lower with Fast and Balanced. Use a setting that gives an acceptable accuracy.
- **The recommended trigger level**: Triggering at level 1 vs level 0 yields 5% - 15% higher throughput (3,795 pages/hr vs. 3360 pages/hr). This is due to the fact that at level 1 the module has less communication with the InputAccel Server since each instance of the module is processing a document task instead of page level tasks.
- **Resolution**: Processing images with lower resolution improves module throughput by 35% when the recognition engine accuracy is set to Accurate.
- **CPU intensive**: NuanceOCR is CPU intensive. Fast CPUs are recommended.
- **Multiple instances on the same machine:** On a multi-core machine, deploy as many NuanceOCR instances as cores. Each additional instance of NuanceOCR running on an additional core yields about 92% of the efficiency that a single, standalone NuanceOCR would do. On a single-core machine it is recommended to deploy only one instance of NuanceOCR. For a single instance of NuanceOCR, adding additional cores will not increase throughput.

## Non-Critical Factors

- **Reporting**: Enabling or disabling reporting is not significant for throughput.
- **Machine memory**: When PDF files are not generated, NuanceOCR is not a memory intensive module; 80 to 100 MB of memory usage is typical. But when PDF files are generated, NuanceOCR may use much more memory; users must follow the recommended memory requirements for client machines provided in the *Release Notes*.
- Add an additional 256 MB RAM for each additional instance of NuanceOCR that will run on a multi-core machine.
- If RAM requirements exceed 3.5 GB, use a 64-bit operating system.
- **Disk usage**: Disk usage is a minor factor since the module releases the disk space after finishing a task. 50 GB free disk space is recommended.
- **Network latency**: Network latency does not significantly impact module performance since data is pre-sent with the task to the module, but even so it is not recommended to run NuanceOCR or any unattended client module over the WAN.

## Sizing Recommendations

A single CPU machine and one instance of NuanceOCR can process 3,360 -6,895 pages per hour, depending on the OCR engine's Accuracy setting when generating full page with PDF. With 2 instances running on 2 CPU cores, throughput is 106% to 195% of one instance running on one CPU core.

## East Euro / APAC OCR: Performance Comparison with NuanceOCR

East Euro / APAC OCR is a client module that performs optical character recognition in multiple languages, including languages of Eastern Europe and Asia Pacific. East Euro / APAC OCR and

NuanceOCR were benchmarked and results were compared. East Euro / APAC OCR has slower throughput than NuanceOCR.

### East Euro / APAC OCR Throughput Compared with NuanceOCR Throughput:

- For full-page OCR (without PDF output):
  - Accurate mode: 33% of the throughput of General-Use OCR/ICR. For example, if General-Use OCR/ICR processes 10,000 pages/hr, East Euro / APAC OCR processes 3333 pages/hr.
  - Balanced and Fast mode: 12% of the throughput of General-Use OCR/ICR.
- For zonal OCR, the difference in throughput depends on number and type of zones.
  - Accurate mode, 5 zones: 33% - 50% of the throughput of General-Use OCR/ICR.
  - Balanced and Fast mode, 5 zones: 20% - 33% of the throughput of General-Use OCR/ICR.
- Adding PDF output to these scenarios adds about 10% processing time to East Euro / APAC OCR performing Full Page OCR and about 15-20% processing time to East Euro / APAC OCR performing Zonal OCR.

# ODBC Export

ODBC Export uses Open Database Connectivity (ODBC) to transfer data between Captiva Capture and various ODBC data source tables. ODBC Export also stores configuration information (such as data sources, actions, and SQL statements) in mappings and uses the connected data source to execute SQL statements.

## Benchmark Results

Refer to *Test Environment Used for Testing of Classification and* Extraction Modules

System Properties for Capture Client Modules and Administrator

| Item | Required |
|---|---|
| Hardware | Lenovo C30 Workstation<br>• Intel Xeon E5-2609 v2 @ 2.5 GHz (quad core)<br>• 4 virtual CPUs allocated to VM<br>• 4 GB RAM allocated to VM<br>• 1 Gbit Network interface card |
| Operating System | vSphere ESXi 5.5 as virtual machine host<br>VM image built with Windows Server 2012 R2 |

Explanation of Columns in Client Module Benchmark Results for explanation of some of the columns in the benchmark results tables.

Table 25. One Instance of ODBC Export Running on a Dual-Core Machine

| Scenario | Through-put (Pages per Hour) | ODBC Export % Process or Time | Aver-age CPU % Uti-lization for Both Cores | Processo r 0 % Uti-lization | Processo r 1 % Uti-lization | Number of IA Value Requests (per second) from Server | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Read / Write Bytes per Second (KB) |
|---|---|---|---|---|---|---|---|---|---|
| 100 fields, 20 Characters, trigger level 1 | 65,729 | 50% | 27% | 35% | 19% | 10,000 | 13 | 556 / 393 | 0 / 29 |
| 100 fields, 20 Characters, trigger level 7 | 100,840 | 31% | 19% | 18% | 19% | 1,000 | 13 | 667 / 597 | 0 / 12 |
| 10 fields, 20 Characters, trigger level 1 | 920,716 | | 18% | 24% | 13% | 1,000 | 13 | 221 / 125 | 0 / 30 |
| 10 fields, 20 Characters, trigger level 7 | 1,428,571 | | 10% | 10% | 11% | 1,000 | 13 | 207 / 217 | 0 / 17 |

Table 26. Two Instances of ODBC Export Running on a Dual-Core Machine

| Scenario | Through-put of 1 Instance (Pages per Hour) | Through-put of Both Instances (Pages per Hour) | Through-put In-crease of 2 Instances Relative to 1 | ODBC Export 1 % Processor Time | ODBC Export 2 % Processor Time | Average CPU % Utilization for Both Instances for Both Cores | Processor 0 % Utilization | Processor 1 % Utilization |
|---|---|---|---|---|---|---|---|---|
| 100 fields, 20 Characters, trigger level 1 | 65,729 | 98,738 | 50% | 46% | 46% | 52% | 50% | 54% |
| 100 fields, 20 Characters, trigger level 7 | 100,840 | 155,575 | 54% | 25% | 26% | 31% | 32% | 29% |

Refer to *Test Environment Used for Testing of ODBC Export* for details of the hardware and software used to gather these benchmark results and *Image Sets and Settings Used › ODBC Export* for the image sets used for the benchmark testing.

## Critical Factors Affecting ODBC Export Performance

- **Number of values exported**: Export only the IA values required by your business needs as exporting fewer IA values improves performance, especially when triggered at level 1. Bytes transmitted between the InputAccel Server and ODBC Export is also reduced when fewer IA values are exported.
- **Trigger level**: Performance improves significantly when triggered at level 7 instead of level 1 because fewer requests are made to InputAccel Server to retrieve data for level 7. The number of bytes transmitted between the InputAccel Server and ODBC Export is also reduced when triggered a level 7.
- **CPU cores**: It is recommended to use at least a dual-core machine, even though ODBC Export is single threaded.
- **Multiple instances on the same machine**: On a multi-core machine it is recommended to deploy as many ODBC Export modules as cores. Since ODBC Export is not CPU bound, it may be possible to run multiple instances on a single-core machine, although this was not benchmarked. Deploying a second instance of ODBC Export on a dual core machine yields a 50% increase in throughput over one instance. Three and four instances on a quad-core machine were not benchmarked, but it is reasonable to expect that the third and fourth instances will each achieve additional throughput of at least 25% of the first instance.
- **Network Latency**: Latency between ODBC Export and the InputAccel Server should be as low as possible. High latency will degrade performance especially when triggered at level 1 due to the large number of requests sent to the InputAccel Server. When triggered at level 1, it is recommended that ODBC Export and the InputAccel Server be on the same LAN. Latency and bandwidth between the ODBC Export module and the database also affects throughput, but this was not benchmarked.

## Non-Critical Factors

- **Reporting**: Enabling or disabling reporting is not significant for throughput.
- **CPU speed**: ODBC Export is not CPU intensive. Even so, reasonably fast CPUs are recommended.
- **Machine Memory**: Memory usage is not significant. Typically less than 20 MB is used. A machine with at least 500 MB RAM is recommended.
- **Disk usage**: Disk usage is an insignificant factor. 50 GB free disk space is recommended, although typical usage is less than 1 GB.

## Sizing Recommendations

A dual core machine with one instance of ODBC Export triggered at level 1 can typically process more than 60,000 pages per hour, substantially more when triggered at level 7 and when fewer values are exported. A second instance of ODBC Export deployed on a dual-core machine when triggered at level 1 increases throughput by 50%.

### To Estimate the Number of Dual-Core Machines and Number of ODBC Export Instances Required

- Estimate the number of pages per hour you will be processing.
- Estimate the number of IA Values to export per page.
- Select the benchmark result that most closely matches your usage, and then use the ODBC Export sizing formula.

### Sizing Formula

**Throughput** = benchmark results * 75%

**Note:** This formula uses only 75% of the benchmark result to normalize the benchmark results to a typical production environment since production environments rarely achieve the results of a controlled benchmark environment.

### Example of Sizing Based on Environment

You have the following processing requirement and want to calculate the number of ODBC Export modules you require to meet your processing needs:

- 130,000 pages per hour throughput requirement
- Triggering at level 1
- 100 IA Values to export per page

Using the formula, calculate the expected throughput of your machine(s) that will run ODBC Export:

- **Throughput (single instance)**: 65,729 * 75% = 49,297
- **Throughput on a dual-core machine with two instances deployed**: 49,297 * 150% = 73,946 (50% increase due to the second instance)
- **Number of dual-core machines that you will need**: Pages to process/dual-core throughput = 130,000 / 73,946 = 1.8 (round up to 2)
- **Number of ODBC Export instances**: 2 * 2 = 4 (2 instances on each machine)

## Production Auto-Learning

Production Auto-Learning (PAL) is an automatic learning process that creates new templates with the images processed in production. Production Auto-Learning has three components as follows:

- Supervisor: manages automatic learning, template creation, placement of index and table fields.
- Collector: collects and stores documents for learning.
- A storage folder: a place where collected documents reside until they are learned or purged. PAL is primarily meant to speed time to production. To increase classification and extraction rates post-learning, review and fine-tune templates in Recognition Designer, per the recommendations found in this document. For a more complete description of PAL features and functionality, refer to the topic *Understanding Production Auto-Learning* in the *Captiva Designer Guide*.

Quick-start instructions on running PAL are provided in the *Understanding PAL High-level Steps* section of the *Captiva Designer Guide*. We recommend going through this document section first when deciding whether or not to implement PAL in your project, and then proceed to the *Captiva Designer Guide* for more general information on PAL.

This section provides system administrators with important considerations and guidelines for implementing, optimizing, and maintaining PAL on a Captiva Capture 7.5 system. These recommendations and guidelines are based on a typical installation scenario with all PAL components deployed on the same machine. Because each installation is unique, these guidelines and recommendations do not apply to every scenario and configuration. Results may vary for your specific production environment. Use the guidelines presented in this section as a starting point to help determine the appropriate settings and expected performance of PAL for your specific production environment.

## Recommendation and Success Factors

When deciding whether to implement PAL, there are a number of variables that must be evaluated to determine the suitability of PAL for meeting specific business needs. The following sections provide information, recommendations, guidance, and critical factors that determine the success of a PAL implementation. Follow these guidelines closely to achieve optimal accuracy and performance with PAL.

Consider these factors individually and together. No single factor determines the success of a PAL implementation. Determining the potential value of PAL to a project requires evaluating all the factors taken together, and weighing the value of time saved in configuration and design against limitations in extraction and classification accuracy. Although the accuracy of PAL created templates can be good, they are not as accurate as manually created templates.

**Note:** It is highly recommended that you implement and thoroughly exercise PAL performance in a test environment before updating any production environment. Based on internal testing, we recommend that you pay particular attention to variables such as: Classification rate, extraction accuracy, Collector growth, template growth, and Captiva Identification module start up times.

It is highly beneficial and recommended to spend a few minutes reviewing and making minor adjustments to PAL templates to achieve maximum extraction accuracy, prior to overwriting your production project with the PAL created project. Over time, as the Project Designer becomes more familiar and comfortable creating PAL templates, the Project Designer can set PAL to automatically send the project to production without any Project Designer review.

The testing on which these guidelines are based was performed in a controlled test environment. These results may vary for your specific production environment. Use the guidelines presented in this document as a starting point to help determine the appropriate settings and expected performance of PAL for your specific production environment.

Consider the following factors when evaluating PAL:

- When is PAL Recommended?
- When is PAL Not Recommended?
- Project Environment
- Nature of the Documents
- Quality and Variability of Images
- Number of Variations Per Document Type
- Number of Pages Per Variation
- Nature of Fields

- Number of Fields to Extract

## When is PAL Recommended?

PAL is a useful tool when implemented thoughtfully in appropriate environments. When deciding whether to implement PAL, the following criteria and scenarios should be considered together to gauge usefulness of PAL for your application.

- In the absence of training images and Project Designers, PAL can automate template creation and field placement.

- Documents are for a new project, so there are no existing graphical templates or free form rules.

- Documents are for an existing project that has poor free form rules or many variations, as PAL can progressively create suitable templates to improve the project.

- Document volume is less than 100,000 images per day. This ensures learning system stability and learning within a single 24-hour period.

- Documents are semi-structured and structured, for example, invoices and forms.

- Less than 5,000 document types or total variations across all document types. PAL is limited to 7,000 variations per project.

A document type is a general description of a document class, for example tax form or invoice. A variation is a known graphical difference between documents of a similar type. For example, variations in document source can result in graphical variations, such as a faxed 2010 tax form versus a 2010 tax form from an MFP. Variations may be intentional, such as a 2009 tax form versus a 2010 tax form, or an “ACME” invoice versus an “ABC Company” invoice.

### Example 4-1. Calculating Total Variations

Consider that the a total of 5000 variations can result from:

- One document type with 10-5,000 variations

- Ten document types with 10-500 variations

- One hundred document types with 10-50 variations

Multiply the number of document types by the expected number of variations to get the total number of variations across all document types.

- The system processes more than 10 variations within a document type and more than 5 document types are received per year.
- The project contains more than 25 fields to extract across all variations.

### Example 4-2. Calculating Fields to Extract

- Ten document types multiplied by ten variations produces 100 templates with 4 fields on each, resulting in 400 fields to place. This is a good scenario for PAL.

- With only 2 document types but 2,500 variations each, 5,000 templates with 2 fields each results in 10,000 fields to position. This is a good scenario for PAL.

The testing on which these guidelines are based was performed in a controlled test environment. These results may vary for your specific production environment. Use the guidelines presented in this document as a starting point to help determine the appropriate settings and expected performance of PAL for your specific production environment.

**Is PAL Recommended for New Projects?** — New projects can benefit from the speed of a PAL implementation, automating new project creation to speed time to production. Then, adjustments and fine-tuning can improve the accuracy of the PAL implementation. Review the criteria listed in the section When is PAL Recommended to determine whether a PAL implementation is appropriate.

**Is PAL Recommended for Existing Projects?** — PAL for existing projects can:

- Automate the classification and extraction of new graphical variations, or graphical templates, that existing projects cannot handle, or cannot handle well. This saves Project Designer time in creating templates for these new graphical variations.
- Increase document throughput for existing projects containing keyword or free form templates. However, free form accuracy may be adversely affected depending on effectiveness of existing free form rules.
- Capture data from documents in languages not currently supported by existing free form rules.

## When is PAL Not Recommended?

PAL is not appropriate for all installations and business scenarios. Using PAL is not appropriate when the system must learn:

- A document the same day it is received. Learning is usually performed nightly.
- More than 100,000 images per day.
- Unstructured documents, such as correspondence or fully handwritten documents, where all documents are different.
- Documents with many multiple choice checkboxes comprising most of the form.
- More than 5,000 document types or total variations across all document types. PAL is limited to 7,000 templates per project, after which new document types and variations will not be learned. Additional variations will need to be keyed manually or free form rules used to automate the data extraction. Old or unused templates can also be removed.
- Less than 25 variations across all document types, or less than 10 variations and less than 5 document types received per year.
- Less than 25 fields to extract across all variations.

For example, consider a scenario with only 3 document types and 2 variations for each type. Assuming 3 fields to place on each template, the resulting 6 templates (three document types multiplied by two variations) require positioning a total of only 18 fields. This is not an efficient scenario for PAL. Manually classify and enter field data for these infrequent variations, or manually create templates and zones if justified by the page volume.

## Project Environment

The project environment includes system capabilities, available resources, and business needs. Evaluate each of these criteria.

PAL delivers maximum value when:

- Training images are not available.
- The Project Designer's time is limited.
- Templates must be learned and created daily or less frequently, such as invoices or loan application forms.
  - For example, PAL is not as useful for batch processing of a single document type in a single day. For example, a vendor processing 10,000 tax forms for a customer in a single day would not benefit from PAL.
- The system learns less than 100,000 images per day. If more than 100,000 images must be learned, learning may not complete overnight and system stability could be affected.

## Nature of the Documents

PAL is best suited for semi-structured and structured documents. Semi-structured documents are those documents where the type of data is the same from document to document, but the location of the data on the document differs. Because there is typically high variability within a document type, Captiva Capture can graphically differentiate between them, recognizing that an invoice from "Vendor A" looks different than an invoice from "Vendor B". PAL can create a graphic template for each vendor, or variation, and successfully place fields on each template to extract the required data. Good return on PAL is seen with semi-structured documents, such as invoices, purchase orders, bills, and other documents where the data extracted remains constant, but the documents are graphically varied.

PAL can also learn structured documents, where both data and the location of that data are the same from document to document. PAL value for structured documents, such as forms, is in automatically creating templates for variations of these documents, as well as automatically placing extraction zones for the many fields that a form may have. Variation in both structured and semi-structured documents depends on a number of factors, including:

- The source of the document: Variability may exist as a result of faxing, scanning, copying, or creating a PDF of the document. Two documents from identical originals can have significant graphical variability introduced by the delivery mechanism, or document source. Different scanners may produce different variations, for example, central scans or distributed scans.
- The version of the document: Variability can occur between versions of a document, for example versions of a single tax form published in different tax years.
- PAL is not well suited for unstructured documents, although it can be used to handle mail room documents on a limited basis as described in *Setting N and Purging the Collector*. Unstructured documents are those documents where both the data and the location of the data differ from one document to another, and even from page to page. There is little or no variability within a document type because every document is unique. Because each document looks different, document types cannot be well grouped and recognized.

## Quality and Variability of Images

In most production environments, images come from many different sources delivered in various formats. Consistent, high-quality images enable PAL to produce the most accurate results in both classification and extraction. When evaluating the potential benefit of PAL, consider the source of documents to be processed. Understanding the source and origination process of the documents regularly entering the system enables more accurate estimating of the image quality and amount of image variability to expect.

Variability refers to the graphical differences between images of the same document type. For example, the document type "invoice" includes many images that are graphically distinct. Each graphically distinct invoice is a variation of the invoice document type. Images can be graphically distinct from one another for many reasons. Image variability can be incidental to the document source, such as when scanned images are skewed or inconsistently aligned on the page. Variability can also be purposeful when several variations of a document type are needed, such as versions of the same tax form for different years. Additionally, documents with multiple pages are graphically different depending on the page. The first page, or header page, of an invoice is different and contains different information than subsequent pages. Each unique invoice, or invoice page, represents a variation of the invoice document type. Most of these limitations are addressed by using the new textual Classification method. This method finds the set of words (labels and its value) that appear in the same place in two documents. If the number of matching words are high, then they may belong to the same class. This algorithm takes into account small variations in text (e.g. "Invoice" vs. "Invoice:") and floating fields (e.g. when there is a table on a page, the fields below the table are relative to the bottom of the table). By default PAL always does graphical and textual classification, because that is better use case for customers in terms of improved accuracy.

Image quality is critical to reducing variability and increasing classification rates. Improving poor quality images means that more images can be recognized as graphically similar, to each other or to an existing template, and thus classified. For example, original documents scanned by trained ScanPlus operators would likely be of higher quality, and have less variability, than documents received as web downloads, photocopies, or by fax. The better the image quality, the greater the potential value of PAL. Obtaining high-quality images is a critical step in optimizing the accuracy and performance of PAL.

For additional information on image quality, image variability, and image resolution requirements, refer to *Image Quality and Resolution*.

## Number of Variations Per Document Type

The expected number of variations per document type is an important consideration when deciding whether and when to implement PAL. Variability refers to the graphical differences between images of the same document type.

When there are few variations of a document type, the value of PAL is low because the gains in classification and extraction rates achieved by manually creating templates for these variations outweigh the configuration time savings achieved with PAL. As the number of variations per document type increases, so do the efficiency gains achieved with PAL. The optimum level for balancing efficiency and accuracy is 50 - 5000 variations per document type, per project, based on our estimates. Actual results may vary from those estimates. When the number of variations exceeds 7,000, PAL stops learning new templates.

For additional help understanding how the number of variations per document type affects PAL performance, refer to *Image Quality and Resolution*.

## Number of Pages Per Variation

The number of images, or page volume, per variation processed each year is important when evaluating a PAL implementation, as the number of images per variation per year determines when *N* is reached and learning occurs.

When very few pages are processed for a variation, for example less than 10 pages per variation, PAL may not add value, because often the *N* value is not reached prior to images being purged and learning does not occur in a reasonable time frame. The more the number of pages per variation increases, the sooner the value of PAL will be realized. Optimal performance occurs when the system processes 100-1000 pages per variation, per year, because *N* is almost always reached and learning occurs more frequently. It is highly recommended to fine-tune templates with high page volumes manually prior to overwriting the production project. Accuracy from these templates is key because the variation is seen frequently by the capture system.

For additional help understanding how the number of pages per variation the system processes each year affects PAL learning, refer to *Image Quality and Resolution*.

## Nature of Fields

The nature of the fields to be recognized is a determining factor when deciding whether to implement PAL. PAL is best suited for auto-extraction and learning of machine printed fields, although there are methods whereby PAL can learn hand printed fields, table fields, barcodes, and the location of checkboxes and bubbles. The primary field types are:

- **Machine printed**: PAL is best suited for automatic extraction and learning of machine printed fields, because operators can continue to perform their high-speed data entry duties without having to use the mouse or perform any other unusual input operations.
- **Hand printed**: PAL requires the operator to rubber band each field, in order for PAL to learn it.
- **Table fields**: Operators are required to use the **Table Wizard** in order for PAL to learn the fields in a table.
- **Checkboxes and Bubbles**: PAL can learn the position of a checkbox or bubble, but does not extract values. Because the position is learned, it helps operators find the position of a field when they key values from images, by quickly focusing the operator on the field.
- **Barcodes**: PAL requires the operator to rubber band each field, in order for PAL to learn its position or extract values. Alternatively, the barcode can be read with a Captiva Capture module like ScanPlus.
- **Multi-line Fields**: When fields span multiple lines, like address fields, extraction is especially challenging. PAL offers limited support for multi-line fields. Handle these fields using Key from Image (KFI) or zonal extraction with scripting.

For more information on field types and the impact field type has on learning, refer to *Tuning Automatic Field Placement*.

## Number of Fields to Extract

Generally, the more fields there are to extract the greater the potential value of PAL. With fewer than 5 fields to extract from a document, and a low number of variations per document, the value of PAL is low because the increase in extraction accuracy rates achieved by manually placing zones for these fields outweighs the configuration time savings achieved with PAL.

As the number of fields to place increases, so does the potential benefit of PAL, as configuration time savings begin to outweigh sacrifices in extraction accuracy rates. PAL delivers the most value when there are more than 25 fields to place, across all document variations.

For maximum classification and extraction accuracy, review and fine-tune PAL created templates as described in *Fine-Tuning Recommendations*.

# Installation Recommendations

## Installing PAL Components

PAL is an optional feature, licensed separately, and installed with Captiva Capture. Complete installation instructions are available in the *Installation Guide*.

System requirements are available in the *System Requirements* section of the *Release Notes*.

## How Should PAL Be Deployed?

The following list provides general guidelines and recommendations for deploying the PAL components:

- The Collector module and document storage folder run on the same local network as the InputAccel Server.
    - Each project requires at least one instance of the Collector module and a dedicated document storage folder. Thus, there must be as many instances of Collector as there are projects. Although each Collector instance requires a separate document storage folder, these folders can all reside on the same physical drive.
    - Although it is possible to run multiple instances of Collector for a single project, it is not generally recommended as one instance is sufficient to handle all but the largest project volumes.
    - The size of the document storage folder, where the Collector puts collected documents, is an important performance consideration. In most cases, the size of the storage folder should not exceed 20 GB.
- For a typical deployment, install all PAL components on the same machine, separate from the InputAccel Server and other components. Installing the components on one machine avoids heavy network traffic delays due to the large number of images and image access events.
- For smaller deployments, install all PAL components on the InputAccel Server.
- Run the development project on the machine running PAL. The production project can be elsewhere.
- Supervisor can monitor multiple projects. Supervisor cannot monitor multiple, identical projects.

For example, consider a business model where multiple branch offices use the same project, sending information to a central InputAccel Server. In this example, the problem is solved by monitoring one project, or the work of one branch office, with PAL and performing learning and updates to that project, then pushing the updated project to all the other branch offices.

## Will PAL Learning Affect Production System Performance?

If PAL components are not installed on the InputAccel Server, then learning will not impact production performance. Only updates to the production project file will impact performance. Therefore, schedule updates to the project file (DPP) between processing batches.

In small deployments, installing PAL components on the InputAccel Server as recommended does not significantly impact performance of either PAL or the InputAccel Server.

### Disabling Windows Automatic Reboot

Microsoft Windows 7 has a setting that enables the operating to reboot automatically, often after installing updates. This reboot can interfere with learning. After installing PAL, disable the Microsoft Windows 7 automatic reboot feature. Refer to operating system documentation from Microsoft for help disabling this feature.

## Sizing Document Storage

Document storage size is based on the estimated volume of documents to collect. Correctly sizing and purging the document storage can improve the performance of PAL. To determine an appropriate value:

- Estimate the volume of documents received per year that will be learned.
- Estimate the anticipated classification rate, as collected documents are purged from storage.
- Calculate the total number of unclassified documents collected per year.
- Calculate the average image size, and then double the figure to account for saving the OCR and XML files with the image.
- Determine the purge schedule.
  - If the purge is based on the size of the Collector, then when the maximum size is reached the purge will start.
  - If purge is based on a predefined period of time, the document storage will continue to grow until the purge date is reached and then will begin purging images by date.
  - Once the purge trigger is reached and purging begins, the purge will usually remove around as many images as are newly collected and the Collector size stabilizes.

**Example 4-3. Sizing the Document Storage Folder**

ABC Company processes invoices and attachments and have implemented PAL. PAL is configured to learn invoices, but not attachments, so only invoices will be collected in document storage. This company processes approximately 200,000 pages per year. Of these pages, approximately 75% are invoices and 25% are attachments, so ABC Company expects 150,000 invoice pages to be managed by PAL.

The PAL is set to purge documents yearly, or when a maximum storage size is reached. Of course, PAL will learn images and images are purged after they are learned, so not all of these invoices will be in the collector at the end of the year. Images used to create templates are purged, and images that correspond to a learned template are not collected, so not all 150,000 invoices will be collected.

Next, ABC Company estimates the number of images that PAL will classify and deducts this number from the total number of potential invoices. Use classification rates as the basis for the estimate. For purposes of example, assume 50% of the images are classified, so 50% of the images are collected.

At the end of the year, 150,000 * 50% = 75,000 images collected.

By evaluating some typical images, ABC Company determines that an average invoice is scanned at 200 DPI and is 40 KB, This number depends completely on the input settings and will differ from customer to customer. After calculating an average image size, double that number to account for the OCR and XML files saved with the image. The average image size in this example is 40 KB, therefore, 80 KB is required to save the image, OCR, and XML files. Thus, the estimate for the amount of space to store collected documents for the 1 year purge cycle is 80 KB * 75,000 pages = 6 GB.

The purge setting is the most critical parameter. The purge starts when either the date (one year in this example) or the maximum size is reached, depending on project settings. The size of the document storage must be based on these settings. In described scenario, the estimated document storage is approximately 6 GB.

Since hard disk space is inexpensive, it is highly recommended that a 50% buffer is included in the estimate. Thus, in this scenario the document storage size for the project should be set to 12 GB. The maximum recommended size for the document storage folder is 20 GB. When calculating document storage size, remember that each project requires a separate instance of the Collector and a dedicated document storage folder.

## Sizing Production Auto-Learning (PAL) - Supervisor

Production Auto-Learning Supervisor is a component of PAL that manages automatic learning, template creation, and placement of index and table fields.

### Benchmark Results

Performance tests focused on automatic learning and template creation; tests related to field placement were not performed.

**Table 27. Production Auto-Learning Duration Based on Collector Size**

| Influence of Collector Size | | | |
|---|---|---|---|
| Collector | | PAL Duration (In Hours) | # Created Templates |
| # of Images | Size (MB) | | |
| 5000 | 464 | 0.2 | 97 |
| 10,000 | 928 | 0.4 | 101 |
| 20,000 | 1,846 | 1.3 | 98 |
| 40,000 | 3,712 | 2.8 | 114 |
| 60,000 | 7,424 | 4.2 | 113 |
| 120,000 | 17,848 | 8.5 | 142 |

**Note:** The initial project used is an empty project with a single template.

**Table 28. Production Auto-Learning Duration Based on Project Size**

| Influence of Initial Project Size: Collector = 5000 Images | | |
|---|---|---|
| Initial Project | PAL duration (in hours) | # created templates |

| Influence of Initial Project Size: Collector = 5000 Images | | | |
|---|---|---|---|
| # of Templates | Size | | |
| 1 | 23 KB | 0.2 | 97 |
| 500 | 66 KB | 0.9 | 97 |
| 1,000 | 145 MB | 1.6 | 97 |
| 2,000 | 387 MB | 4.1 | 97 |

Refer to *Test Environment Used for Testing of Client Modules and Administrator* for details of the hardware and software used to gather these benchmark results and *Image Sets and Settings Used ›PAL Supervisor* for the image sets used for the PAL Supervisor benchmark testing.

### Critical Factors Affecting Auto-Learning Supervisor Performance and Tuning

- **Collector size**: When the Collector has more than 20,000 images, Auto-Learning duration is nearly linear.
- **Project size (initial number of templates)**: The more templates there are in the project, the longer it takes to add new templates. For a large project (initial number of templates = 2,000) with small document storage, Auto Learning takes approximately 3 to 4 hours more than with a small project having few templates. For a large project, (initial number of templates = 2,000) with large document storage, Auto Learning takes 7 to 10 hours more than with a small project. For example, for a collector of 60,000 images, Auto Learning takes 4.2 hours for an empty project, while it takes 11.5 hours for a project with 2,000 templates.
- **CPU cores**: Only 1 instance of Supervisor can be run on a single core machine. Supervisor is single threaded and does not display any significant increase in performance when using additional CPU cores.
- **CPU speed**: CPU usage is significant (between 20% and 40%). Fast CPUs are recommended.
- **Other recommendations**:
  - Install Collector, Supervisor, and the Document Storage on the same machine. This facilitates faster comparison and faster project updates. Also, locating the Dispatcher project on the Supervisor machine results in faster project updates.
  - **Disk speed**: High speed disk access is important because of the access to data storage during auto learning.

## Setting *N* and Purging the Collector

The *N* value represents the minimum number of documents required to create a template. Fine-tuning *N* and the purge settings is fundamental to improving the efficiency and accuracy of PAL.

The testing on which these guidelines are based was performed in a controlled test environment. These results may vary for your specific production environment. Use the guidelines presented in this document as a starting point to help determine the appropriate settings and expected performance of PAL for your specific production environment.

The following sections provide information, recommendations, and guidance to select PAL settings according to business needs.

## How do *N* and Purge Settings Affect PAL Accuracy and Performance

Defining the value of *N* affects the efficiency and accuracy of PAL.

When defining the *N* value, the image quality as well as the number of document types and variations must be taken into account. Detailed information is available in the *Image Quality and Resolution* section.

Generally speaking, setting up a low *N* improves classification since more variations are represented and consequently more templates are learned. However, a low value of *N* reduces recognition efficiency on classified images. To improve recognition performances in classified images, it is better to define a high value of *N*. Many images are necessary to create a template of high quality. If several fields must be placed on the template it is better to have a template of higher quality overall, especially if field placement varies on the image.

Setting *N* to 4 is a trade-off between getting a higher classification rate with fewer images in the Collector, and still maintaining acceptable extraction accuracy on the classified images — leading to the highest overall extraction. Extraction accuracy on classified images does not start hitting acceptable levels until *N* is at least 4, and they continue to climb until *N* is set to 8 or 10, where extraction accuracy starts declining. Classification rates are optimal when *N* is set to 2 and they start declining significantly, for a given set of images, as *N* increases.

Setting the *N* value also depends on whether or not infrequent templates should be learned. If a low *N* is defined, PAL learns frequent and infrequent templates. PAL performance may be affected because the Collector is overloaded and the learning speed decreases. Frequent templates may not be learned because the project is already full. To prevent this scenario, set a high value of *N* so infrequent templates are not learned, the Collector is not overloaded, and frequent or more valuable templates can be learned. For example, if you are learning document types that only come in once or twice a week, then setting *N* to 8 means that many of those documents will not be learned for months. If the document type comes in dozens of time each day, then setting *N* to 8 will increase accuracy and still have an acceptable classification rate and learning time.

The operator effort should also be taken into account when defining the *N* value. If a low *N* value is defined, such as 2, the operator does not need to validate many images before getting the benefits of learned templates. However, overtime, they will take more fields to correct than if the optimal *N* value of 4 was set.

## Setting the *N* Value

The following section gives some guidance to fine-tune the *N* value for three application types:

- Setting the *N* Value for Invoice Applications
- Setting the *N* Value for Forms Applications
- Setting the *N* Value for Mail Room Applications

### Setting the N Value for Invoice Applications

The nature of the application is a determining factor when defining the *N* value. Because there is high variability within a document type, PAL can graphically differentiate an invoice from "Vendor A" than an invoice from "Vendor B". PAL is best suited when working with invoices because the data extracted remains constant even if documents graphically vary.

Setting the *N* value depends on the volume and the variability of the documents type. This section gives recommendations for setting *N* when working on:

- Small invoice applications
- Medium invoice applications
- Large invoice applications

### Setting the N for Small Invoice Applications

This section gives some examples and recommendations for setting *N* for small invoice applications.

Setting a low *N* enables PAL to learn many templates since the number of document types is quite small. Setting a higher value of *N* would affect the potential benefits of PAL. Few document types would be learned and the benefit of PAL would occur very late in the project.

If *N* is too low, for example *N* = 2, templates created by PAL are less accurate than templates created with free form rules. The aim of PAL is to get better results than free form with lower human cost.

For classification and data extraction, setting *N* to the default value of 4 is a good trade-off between having a high classification rate and a good extraction accuracy on the classified invoices. For textual classification only, the recommended *N* value is 3. Setting a higher *N* enables PAL to get better accuracy for data extraction but it would decrease classification rate because there would be not enough document types in the Collector to be learned.

**Table 29. Setting *N* Value for Small Invoice Applications**

| | Volume of documents per year | Volume of documents per day | Number of document types | Number of fields to be placed | Recommended value of N for classification and data extraction | | | Purge of the Collector |
|---|---|---|---|---|---|---|---|---|
| | | | | | Default (for Textual and Graphical Classification) | Textual only | Graphical only | |
| Small Invoice Applications | 25,000 | 100 | 1,000 to 5,000 | 5 to 20 index fields<br>5 to 10 table fields | 4 | 3 | 4 | Once a year |

### Setting the N for Medium Invoice Applications

This section gives some examples and recommendations for setting N for medium invoice applications.

Setting up the value of *N* for medium invoice applications is almost the same as for small invoice applications.

*N* can be a bit higher than for small volumes since the number of variations is greater. This setting enables PAL to filter infrequent templates. Only valuable or the most representative templates should be learned.

Templates created by PAL must have better results than templates created with free form rules. We recommend setting *N* to 4 to get better accuracy.

The correct balance between classification and data extraction is higher for medium than for small volumes. For both *N* should be set to 6 so that infrequent templates are not learned by PAL and there is no risk they will be learned in place of more representative templates.

An alternative strategy would be to create templates for the highest volume of document types manually to achieve better accuracy than PAL can provide. Then, set *N* to 5 (default) to learn the remaining, less frequent, document types. In case of textual classification only, the recommended *N* setting is 4.

**Table 30. Setting *N* Value for Medium Invoice Applications**

| | Volume of documents per year | Volume of documents per day | Number of document types | Number of fields to be placed | Recommended value of N for classification and data extraction | | | Purge of the Collector |
|---|---|---|---|---|---|---|---|---|
| | | | | | Default (for Textual and Graphical Classification) | Textual only | Graphical only | |
| Medium Invoice Applications | 400,000 | 2,000 | 10,000 to 30,000 | 5 to 20 index fields<br>5 to 10 table fields | 5 | 4 | 5 | Every 6 months |

### Setting the N for Large Invoice Applications

This section gives some examples and recommendations for setting *N* for large invoice applications.

This recommendation is almost the same as for medium invoice application except that the value of *N* for classification is higher. Since the number of document types is even more important in large invoice applications, infrequent templates should be skipped so PAL performance and accuracy are not affected. Otherwise, the Collector is quickly overloaded and frequent or more valuable templates cannot be learned. Again, setting a higher value of *N* prevents learning of infrequent templates.

One strategy, would be to set *N* to the default value of 8, initially for a short period, such as a month or two, and let the system learn the highest volume of document types with the highest accuracy of extraction that PAL can provide. For textual classification only, the recommended *N* value would be 6. Then, *N* can be reduced over the course of several more weeks down to 4, to learn the less frequent document types.

**Table 31. Setting *N* Value for Large Invoice Applications**

| | Volume of documents per year | Volume of documents per day | Number of document types | Number of fields to be placed | Recommended value of N for classification and data extraction | | | Purge of the Collector |
|---|---|---|---|---|---|---|---|---|
| | | | | | Default (for Textual and Graphical Classification) | Textual only | Graphical only | |
| Large Invoice Applications | 1M to 4M | 5,000 to 20,000 | 10,000 to 30,000 | 5 to 20 index fields<br>5 to 10 table fields | 8 | 6 | 8 | Every 4 months if PAL learns 1 million pages per year .<br>Every 2 months if PAL learns 4 million pages per year. |

### Setting the N Value for Forms Applications

PAL can be used in forms applications to learn machine-printed and handwritten forms. This section gives some examples and recommendations for setting N for forms applications.

- For machine-printed forms: Since forms have many document types, setting a higher *N* does not affect the classification rate unless the image quality is poor. For data extraction, *N* should be set to a high value since templates of high accuracy are required. We highly recommend manually fine-tuning templates frequently used to get the best accuracy. Accuracy from these templates is key because the variation is used frequently.
- A good trade-off between classification and data extraction is to set *N* to 6 to get good accuracy for extraction and good learning time for classification. The aim is to reduce time and work for the operator by eliminating the need to validate too many fields manually.
- For handwritten forms: PAL is not able to locate handwritten fields on documents so it relies on rubber band coordinates. If the operator rubber bands the zone where the data must be extracted, PAL can provide the same rubber band accuracy for several documents. Setting a low *N* is recommended when rubber band is used a lot, for example *N* = 2, in order to reduce the operator effort.

**Table 32. Setting *N* Value for Machine-printed and Handwritten Forms**

| | Volume of documents per year | Volume of documents per day | Number of document types | Number of fields to be placed | Recommended value of N for classification and data extraction | | | Purge of the Collector |
|---|---|---|---|---|---|---|---|---|
| | | | | | Default (for Textual and Graphical Classification) | Textual only | Graphical only | |
| Machine-printed Forms Applications | 1M to 10M | 5,000 to 50,000 | 20 to 50 per documents | 10 to 200 index fields<br>0 to 20 table fields | 6 | 6 | 6 | Once a month |
| Handwritten Forms Applications | 1M to 10M | 5,000 to 50,000 | 20 to 50 per documents | 10 to 200 index fields<br>0 to 20 table fields | 2 | 2 | 2 | Once a month |

### Setting the N Value for Mail Room Applications

PAL is not recommended for mail room applications with unknown document types. This section gives some examples and recommendations for setting N for mail room applications.

Main criteria for using PAL in mail room applications: If there are many document types with many variations per document, PAL may not learn them all. Some of them are more important, so PAL filters the document types to learn which are the most representative. N should then be set to 8 (default), and in case of textual classification only to 6, to skip infrequent document types and variations. PAL is only valuable in mail room applications if there are frequent document types and variations.

For classification, N can be low (N = 2) since there are few document types or variations per volume. For data extraction, N is particularly hard to define because it depends on several potential use cases:

- If free form rules are defined in the project: To help ensure that PAL accuracy is better than free form accuracy, setting *N* to 4 is recommended.
- If handwritten fields require rubber band: PAL can learn images quickly when rubber band is used. Accuracy after learning 2 or 3 images is similar to learning 10 images without rubber band because an operator performs the rubber banding, which enhances accuracy. Fewer images are thus required to find the best zone. Additionally, rubber banding images is time-consuming, although precise. Therefore, *N* should be low value (*N* = 2).

A good trade-off would be setting N to 4 or 6 depending on the number of variations. If variation is significant, we only want to learn frequent variations. To prevent learning infrequent variations it is recommended to set a higher N.

**Table 33. Setting *N* Value for Large Mail Room Applications**

| | Volume of documents per year | Volume of documents per day | Number of document types | Number of fields to be placed | Recommended value of N for classification and data extraction | | | Purge of the Collector |
|---|---|---|---|---|---|---|---|---|
| | | | | | Default (for Textual and Graphical Classification) | Textual only | Graphical only | |
| Large Mail room Applications | 1M to 4M | 50,000 to 500,000 | 30,000 | 1 to 5 machine-printed or handwritten index fields | 8 | 6 | 8 | Every 2 months if PAL learns 1 million pages per year.<br>Every 4 months if PAL learns 4 million pages per year. |

## Running PAL in Production

It is highly recommended that you implement and thoroughly exercise PAL performance in a test environment before updating any production environment. Based on internal testing, we recommend that you pay particular attention to variables such as: Classification rate, extraction accuracy, Collector growth, template growth, and Captiva Identification and Completion module start up times.

Once PAL is configured and running in production, there are optional maintenance tasks to keep accuracy and performance as high as possible. Before overwriting your production project with a PAL created project, it is recommended to manually review the project and fine-tune application settings and templates to maximize PAL accuracy and performance as described in this guide. Templates can also be sent automatically to production, fine-tuned, and then updated on the production server. Information on automatically updating the project with learned templates is available in the *Selecting Production Auto-Learning Project Options* section of the *Captiva Capture Guide*.

Because template fine-tuning is an ongoing task, as new templates enter the system and are learned, running PAL in production involves tasks already discussed elsewhere in this Guide. This section provides a simple overview of those tasks in the context of a production workflow including:

- Using Rubber Band Recognition
- Managing Template Learning
- Maintaining the Collector and Document Storage

### Using Rubber Band Recognition

Rubber band recognition is used when handling hand printed characters, checkboxes, bubbles, and barcodes.PAL requires the operator to rubber band each field, in order for PAL to learn it. Rubber band can also be used for machine printed fields if the operator so chooses, but this is not required for PAL to learn them.

- Hand printed fields and barcodes can be placed and data extraction performed using rubber band recognition. Placing fields on a few images using rubber band enables PAL to place anchors and fields on subsequent images.
- PAL uses rubber band to learn the position of checkboxes and bubbles, although these cannot be extracted. Learning the position enables PAL to help the operator to see where to look on the screen when they key from image or rubber band a checkbox.

Refer to *Using Rubber Band Recognition* in the *Captiva Capture Guide* for help understanding and using rubber band technology.

### Managing Template Learning

The ongoing learning of templates is the primary job of PAL and reviewing template learning a critical part of the Project Designer's role, especially with new variations entering the workflow with which the Project Designer is not familiar.

- **Reviewing templates**: Although templates can be automatically sent to production and never reviewed, it is highly recommended that templates are reviewed and fine-tuned. Benefits of reviewing templates are that problems are identified and the templates improved, providing greater extraction accuracy.

- **Deploying templates**: Templates can be sent automatically to production, or saved locally for review, fine-tuning, and manual deployment.

To send templates to production automatically, select the **Send automatically to production** option from the **Production Auto-Learning** tab of the **Project Options** window in **Dispatcher Manager**. It is recommended that this option remain disabled until after initial learning and fine-tuning of templates is complete. Find specific guidelines for fine-tuning templates in the section *Fine-Tuning Recommendations*.

- **Updating learned templates**: Once templates are learned, they are not automatically updated over time. If a field is not placed, or placed inaccurately when the template is learned, it will remain so unless the template is manually updated and the field properly placed. The fine-tuning guidelines in this guide provide information about how to fine-tune and update templates once they are learned. When updating a learned template, remember that it must be copied to the production server and the updated project templates reloaded on client machines.
- **Scheduling learning**: Learning usually runs every night when PAL is first implemented for a project. For efficiency, the learning schedule can be adjusted over time as described in *Supervisor Recommendations*.

Additional help and general recommendations are in the *Managing Templates* section of the *Captiva Capture Guide*.

### Maintaining the Collector and Document Storage

Each project has both an instance of the Collector and a document storage folder, both of which require thoughtful setup and maintenance. As discussed in the section *Setting N and Purging the Collector*, the setting of N and purge settings for the Collector have a direct impact on how PAL learns templates, classification and extraction accuracy, and performance.

Monitor document storage and template learning, making adjustments to learning frequency and N values based on the number of documents collected and the number successfully classified. Over time, as the template base grows, project needs will change and careful monitoring and analysis can provide valuable information for further fine-tuning and optimizing templates and learning.

Information on sizing document storage hardware is available in this guide.

## Fine-Tuning Recommendations

Initially, configuring PAL not to auto-update the production project is advised, to give the Project Designer an opportunity to fine-tune templates created by PAL before overwriting the production project with the PAL created project. This fine-tuning maximizes classification and extraction accuracy rates. Fine-tuning a PAL template includes improving image quality, adjusting recognition zones, adjusting application settings, and customizing field OCR settings. Even minor tuning, such as adjusting the recognition zones that PAL created for each field, can result in significant improvements in accuracy, so time invested fine-tuning templates is generally offset by accuracy gains.

The recommendations and guidelines provided in this section support fine-tuning PAL created templates. Implementing these recommendations at the start of a project, and as new templates are created by PAL in production, will maximize PAL accuracy and performance. Topics in this section include:

- Image Quality and Resolution

- Tuning Application Settings
- Tuning PAL Created Templates for Classification
- Tuning Automatic Field Placement
- Supervisor Recommendations

## Image Quality and Resolution

Improving image quality, understanding and managing image variability, and complying with image resolution requirements are critical design time factors. Variability refers to the graphical differences between images of the same document type. Refer to *Quality and Variability of Images* for more discussion of variability.

### Image Quality

Improving image quality is one of the most critical and straightforward ways to improve PAL accuracy and performance. As image quality degrades, variability increases and PAL accuracy decreases. Thus, one important outcome of ensuring high image quality is that there is little variability introduced by the image capture process. Use images of the highest possible quality for maximum efficiency and accuracy with PAL.

### Image Resolution Requirements

For PAL to collect images, all images being learned must be the same resolution as the project. Images with a resolution different than the project resolution are not collected. When creating the generic template, it is critical to select a sample image that is representative of the images processed in production. The resolution of the sample image selected determines the project resolution.

## Tuning Application Settings

This section describes adjustments to design-time settings that can help improve PAL accuracy.

- Tuning Index Family Settings
- Tuning Free Form Settings
- Tuning OCR Settings

### Tuning Index Family Settings

There are several index family settings dedicated to PAL. These settings include the ability to enable or disable learning on a field, and some data formatting options. For PAL to learn a field, the Learned in production option must be True. Otherwise, the field is not learned.

Settings for these fields are project-specific so there are no general fine-tuning recommendations. For help setting the Culture, Data type, and Keyed format options, refer to the section *Selecting Index and Table Field Properties* in the *Captiva Capture Guide*.

**Note:** PAL cannot automatically apply filters defined on the Image Clean Up tab of the Index Family Editor Field Properties panel. To apply image filters, manually fine-tune the template after learning.

#### Tuning InputAccel for Invoices Index Family Editor Settings

This section provides information specific to InputAccel for Invoices field settings related to PAL. These settings apply to out-of-box fields and are set on the Field Properties panel in the Index Family Editor. PAL is turned on or off at the field level by changing the Learned in Production value from True to False. In most cases where turning PAL off is recommended, it is because free form rules are more accurate and thus recommended.

When learning is disabled on a field because free form rules are recommended, PAL must be configured to place fields using free form rules. Thus, in addition to setting the Learned in Production value to false, it is necessary to set project options to enable PAL to apply free form rules as described in *Tuning Application Settings › Tuning Free Form Settings*.

As a general rule, PAL excels at locating fields that are on most of the documents used to create a template. If a particular field is often missing from many of the documents, it may be preferable to use free form rules for the field. Follow these recommendations for handling critical fields:

- Invoice Credit: Turn off PAL for the field and enable free form settings, as they are more accurate than PAL for handling this field. The probability of PAL falsely identifying a document as an invoice or failing to classify a credit memo correctly is high, creating a risk of false positives. Thus, always use free form rules to handle the Invoice Credit field.
- Invoice Shipping: Turn off PAL for the field and enable free form settings, as they are generally more accurate than PAL for handling this field. However, if most invoices entering the system contain a shipping amount, it may be worth turning PAL on to test if accuracy can be improved with PAL over free form rules.
- Invoice Tax Amount: Turn off PAL for the field and enable free form settings, as they are generally more accurate than PAL for handling this field. However, if most invoices entering the system contain a tax amount, it may be worth turning PAL on to test if accuracy can be improved with PAL over free form rules.
- Invoice Date: Turn off PAL for the field and enable free form settings, unless a Project Designer reviews and tunes PAL templates periodically during production. Carefully fine-tuning a PAL template could produce more accurate results than free form rules, but without this fine-tuning it is better to disable PAL and apply free form rules.

As with all recommendations in this guide, these guidelines come from test results based on an internal image set for a “typical” customer. Since every installation is different, results may vary based on factors specific to the implementation, such as invoice quality and variety. The testing on which these guidelines are based was performed in a controlled test environment. These results may vary for your specific production environment. Use the guidelines presented in this document as a starting point to help determine the appropriate settings and expected performance of PAL for your specific production environment.

### Tuning Free Form Settings

Use the recommendations in this section to decide when to activate the Collector and reapply free form rules. The Collector can be activated from the Project Options window Production Auto-Learning tab. The option Apply Free Form settings on templates when index field learning fails on this tab controls when existing free form rules are applied to a project if PAL fails to place a field. PAL does not use free form rules directly, but can default to these rules based on this setting. If no free form rules are defined, this option is ignored, even if it is enabled for the project.

PAL does not start collecting documents until the Collector is activated. Then, if PAL cannot place the field zone on the PAL created template, free form rules exist, and the option to apply free form settings is selected, then the field is placed using defined free form rules. This is how PAL reverts to free form if free form rules exist.

If PAL cannot place the field and free form rules are defined and enabled, the field is placed to cover the full page, DFT settings are applied, and OCR properties are imported. If neither the PAL option nor the free form option is selected, then the field is not placed and then not read in production. If this template is classified in production, no value will be extracted for this field, and the operator must key from image.

The option to apply free form rules is selected by default and it is recommended that this option remain enabled if working free form rules exist. Use the setting when you have existing free form rules that provide better accuracy than can be achieved without free form rules.

**Note:** Remember that when free form rules are applied, the field zone is placed to cover the full page and requires full page OCR, so classification speed can be affected. For classification projects only, disable the Apply Free Form settings on templates when index field learning fails option to improve classification speed.

### Tuning OCR Settings

For PAL to work, one keyword rule or free form setting is required to create the OCR file used by PAL learning. When this rule is created, an OCR engine is selected.

When a generic template is created, the OCR, ICR, OMR, or barcode recognition engine defined for the field in the index family is used when placing fields. Otherwise, OCR results from the generic template are used, if available.

It is highly recommended to fine-tune the OCR engine for each field in the index family. Selecting an appropriate engine to handle amounts, dates, numbers, hand printed text, and other expected values will greatly improve the likelihood that fields will be placed and recognized accurately. Selecting a generic full text engine significantly reduces the likelihood of accurate results. Refer to the *Understanding Recognition Engines and Engine Configuration Files* section of the *Captiva Capture Guide* for help selecting appropriate recognition engines.

## Tuning PAL Created Templates for Classification

PAL is primarily meant to speed time to production. To increase classification rates post-production, review PAL created templates with a focus on keyword rules and high precision anchors (HPA) for improving document classification. Consider adding keyword and HPA templates to the project to handle the most critical document types and achieve the highest possible classification rates.

### Adding Keyword Rules

Adding keyword rules is an easy and low cost way to improve document classification. Consider adding keyword templates, based on PAL classification results, to improve classification rates. This is particularly useful at the start of a project, when adding simple keyword rules can help PAL start learning templates more quickly. For help understanding and creating keyword rules, refer to the section *Defining Keywords* in the *Captiva Capture Guide*.

### Adding High Precision Anchors (HPA)

Adding HPA templates to the project is another way to improve classification. HPA templates accurately differentiate between documents that have similar graphic layouts because anchors are positioned manually and thus precisely. Such precision in anchor positions makes HPA a flexible method to differentiate documents and improve classification rates for common or critical variations. For help understanding and placing high precision anchors, refer to the section *Understanding HPA Classification* in the *Captiva Capture Guide*.

## Tuning Automatic Field Placement

PAL automatically places fields on documents as part of creating templates. PAL detects anchors and places field by comparing multiple images, forming hypotheses, and refining those hypotheses to determine appropriate anchors and the correct placement of fields. Fine-tuning field placement settings can substantially improve both successful field placement and recognition accuracy.

Field placement is fine-tuned by adjusting recognition zones. Although PAL can effectively place many fields, simple adjustments of zones can noticeably increase extraction rates and is highly recommended. Sometimes PAL creates anchors that are very far away from the extraction zone, as described in *Tuning Automatic Field Placement › Resolving Common Problems*. It is not necessary to adjust anchors. Instead, focus on ensuring the data extraction zone is correctly sized around the data to be extracted.

Additional information on field placement is available in the Captiva Capture Guide:

- Find details on how PAL determines anchor and field placement in the *Understanding the Field Placement Algorithm* section.
- General information and recommendations on field placement with PAL are in the section *Recommendations for Field Placements*.

Topics in this section include:

- Tuning Recommendations for Different Field Types
- Troubleshooting Field Placement

### Tuning Recommendations for Different Field Types

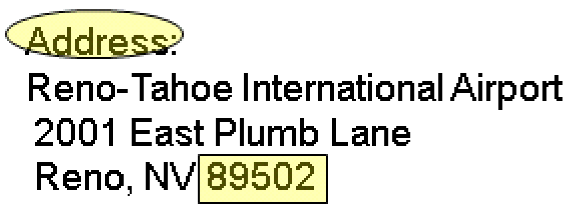
When fine-tuning field placement, consider the nature of the field and how it is handled by PAL. As a general rule, PAL excels at locating fields that appear on most of the documents used to create a template. If a particular field is missing from many documents, it may be preferable to use free form rules or KFI for the field to avoid false positives. For example, if a field that is not on many documents is equal to 0, it is possible that PAL will place the zone around another 0 that is not associated with the field, and then when the field actually appears, PAL finds the incorrect zone. Use recommendations in this section to make appropriate adjustments to the placement of each field type.

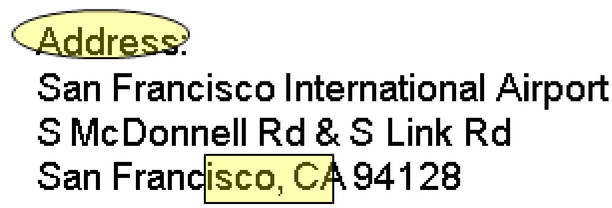
- Machine printed: PAL is best suited for automatic extraction and learning of machine printed fields. Ensuring high-quality images, adjusting field placement, and adding anchors are all ways to improve accuracy with machine printed fields.
- Hand printed: Try the following suggestions for resolving issues handling hand printed fields:
  - If the field is not placed, have an operator place the field using rubber band.
  - If the field is placed but the values are incorrect, have an operator fix the field value without rubber band.
- Checkboxes and bubbles: PAL does not currently learn checkbox fields during extraction, though the position can be learned. PAL can locate the field on the image, learning to zoom based on rubber band values and eliminating the time it takes operators locate and zoom the field in Completion. However, the operator still needs to key the value.
- Barcodes: PAL requires the operator to rubber band each field, in order for PAL to learn its position or extract values. If the barcode is on a separator sheet, an alternatives is

to read the barcode with an Captiva Capture module like ScanPlus. Otherwise, have an operator to rubber band the barcode.

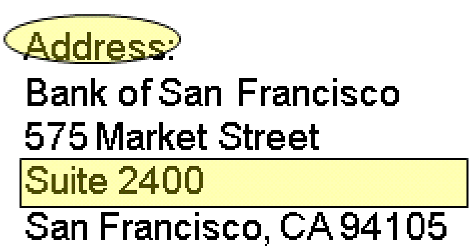
- Multi-line Fields: When fields span multiple lines, like address fields, extraction is especially challenging, even when a template is manually created. Consider the difficulty of extracting the zip code from a standard address. The location of the state and zip code fields in a standard address are dependent on the number of characters in the city.

For PAL to learn address fields, or other fields with more than one line, some additional fine-tuning and manual steps are needed. One solution is to learn the entire city, state, zip code line, rather than trying to learn only the zip code. Then, optionally set field properties to have the address field visible and keyed in Completion. This means setting the field to visible in Completion, not learned by PAL, and with no OCR. Alternatively, the address line can be hidden and the values extracted with OCR and the field learned by PAL. In this case, the extraction must be fixed in Completion, using scripting to identify and extract the city, state, and zip code values.




Unfortunately, this solution does not completely resolve the issue. The number of lines in an address can vary, depending of whether an attention, floor, suite number, or other information is included in the address. This shifts the location of the zip code, and the entire city, state, zip code line.




Although this solution has some limitations, the recommendation to use either key from image or zonal extraction with scripting, produces better results when handling address fields than attempting to extract and learn individual values when working with fields spanning multiple lines.

- Table fields: PAL is less practical for learning tables because it is not fully automated, but PAL can handle table fields based on values defined by the Table Wizard. It is not necessary to use the wizard on every image. Once the wizard has been used on one image in a group, PAL can place the table fields and anchors on subsequent classified images.
- It is important to place table fields carefully, completely covering the column even if data only appears in one row, because table length may be different from document to document for the same variation. When PAL creates the template it should cover all potential rows. Use the Table Wizard following guidelines detailed in the section Understanding the Table Wizard in the Captiva Capture Guide.

## Troubleshooting Field Placement

This section offers some tips for troubleshooting field placement problems, including:

- What if The Field is Not Found or Not Learned?
- Resolving Common Problems

### What if The Field is Not Found or Not Learned?

There are several things to check and try if the field is not found or not learned:

- Verify that the PAL is enabled for the project on the Project Options window Production Auto-Learning tab.
- Adjust the recognition zones created by PAL.
- If free form rules are defined in the project, verify that the Apply Free Form settings on templates when index field learning fails option is enabled on the Project Options window Production Auto-Learning tab. If the PAL cannot place an index field on the template, and free form rules are defined, this option can enable field placement based on free form rules.
- Place the field on the template manually.

### Resolving Common Problems

The following are some common potential issues, and suggestions for resolving them:

- Poorly placed anchors: It is fairly common for PAL to place anchors in locations that seem odd or not intuitive. However, the impact of oddly placed anchors is low. Oddly placed anchors can be working anchors. If the anchor works, no adjustments are necessary. If the anchor does not give satisfactory results, define a more appropriate anchor when fine-tuning the template.
- Poorly placed fields: Field placement may be poor, especially when the value of the field is 0. The impact of this problem depends greatly on the documents in the workflow. If N images have same values, and images in production also have that value, there is no impact. Rely on controls to catch special cases where the values are different. One potential problem of a poorly placed field is that the image zooms to the wrong location during validation in Completion. If the field causes issues during testing, adjust the template as described in *Tuning Automatic Field Placement*.
- Field zone is too small or large: The zone created for the field may be too small or too large, depending on the sample size learned. Generally no action is required, although the zone can be adjusted during template fine-tuning. If the zone is too large, the impact is usually positive, as the field can accommodate longer or larger values on subsequent images. If the field is too small and data is not fully extracted, enlarge the field so it fully covers the data to extract.
- Unexpected OCR results: Sometimes OCR results are not as accurate as expected, although PAL is compatible with all zonal and full page engines available in Dispatcher for InputAccel. PAL accuracy and ability to place keyed fields is highly dependent on full page OCR accuracy. Although trial and error is often the best way to determine OCR engine settings for a specific environment, the Western OCR engine currently provides optimal full page OCR results and is recommended for full page OCR. For field level OCR, the data type drives OCR engine selection, as described in *Tuning Application Settings ›Tuning OCR Settings*.

### Supervisor Recommendations

In most cases, schedule learning to run nightly. Usually, learning starts as soon after the end of the work day as possible, to maximize learning time and minimize the possibility that learning will not finish before the start of the next workday. In cases where learning cannot complete overnight, it may be scheduled to run during the weekend, or during other specified downtimes. Learning can also be scheduled to run every other day.

Appropriate scheduling depends on a number of factors specific to each installation, including the estimated duration of learning. Duration depends on the number of images in the storage folder, the number of generated templates, and the number of fields to place on the templates. Refer to the following resources to understand the Supervisor and how learning is scheduled and managed.

- For more information on scheduling learning, refer to the topics *Recommendations for Scheduling Learning and Selecting Supervisor Settings* in the *Captiva Capture Guide*.
- Detailed information on selecting Supervisor settings is available in the Understanding Supervisor section of the Captiva Capture Guide.
- Performance testing and benchmark information for the Supervisor is available in this guide.

#### When Should Learning Stop?

Immediately following the implementation of PAL, there are many new documents entering the system for PAL to learn. Usually, learning runs nightly and new templates created regularly. Over time, the number of new documents representing images for which no template exists, starts to decrease as the system encounters fewer and fewer unfamiliar documents.

As the number of newly created templates starts to decrease, the value of PAL learning and creating new templates should be evaluated against the cost of learning and learning frequency adjusted accordingly. One cost of learning is that the project must be reloaded on the Completion module each time it is updated. As the number of templates increases, the amount of time it takes to load the templates on the client also increases. If the number of templates is large, reloading can be time-consuming. For example, if the system has 1000 templates and learning creates only one or two new templates, the cost of reloading the templates must be weighed against the value added by the two newly created templates.

Administrators must determine an appropriate learning schedule based on factors such as document volume, document variability, and the frequency with which new documents enter the workflow. If after 6 months of production, new documents are no longer received daily, new templates are created only occasionally, or new templates are similar to existing templates and of little value, adjust the learning schedule. In this example, adjust learning frequency so that learning occurs less often, such as once per week. Disabling the option to send the new templates to production automatically gives an administrator time to assess the value of the new template and fine-tune it for improved accuracy, before deploying it to production.

## Standard Export

The Standard Export module performs data and/or file export from a batch to the specified destination. The data can be exported in several supported file formats.

## Test Scenarios

The tests performed for Standard Export fall into three usage categories: data export, file export, and email export. Each category included unique usage scenarios.

The file export scenarios were used to test the impact of file size and trigger level on the module throughput.

**Table 34. File Export scenarios**

| Scenario | Description |
| --- | --- |
| 1 | Export Doc Level stage files, file size = 1 MB, 100 files per batch, trigger level 1 |
| 2 | Export Doc Level stage files, file size = 10 MB, 100 files per batch, trigger level 1 |
| 3 | Export Doc Level stage files, file size = 100 MB, 100 files per batch, trigger level 1 |
| 4 | Export Doc Level stage files, file size = 1 MB, 100 files per batch, trigger level 7 |
| 5 | Export Doc Level stage files, file size = 10 MB, 100 files per batch, trigger level 7 |
| 6 | Export Doc Level stage files, file size = 100 MB, 100 files per batch, trigger level 7 |

The above scenarios were tested using:

- 100 documents per batch and one file per document (100 files per batch).
- No folder separation (one folder contained all 100 documents).
- The document sizes 1 MB, 10 MB, 100 MB (the worst case batch used 100 documents, 100 MB each, which gives 10,000 MB, or 10 GB).
- Export to the UNC path on a local 1 Gbps LAN (not to a local disk).
- Trigger levels 1 and 7.

The email export scenarios were used to test how the module's throughput is affected by the number of attachments, by the attached file size, and by the latency of the mail server.

**Table 35. Email Export scenarios**

| Scenario | Description |
| --- | --- |
| 1 | Export 100 Folder Level emails with: 1 x 100 KB attachment, trigger level 2 |
| 2 | Export 100 Folder Level emails with: 1 x 1 MB attachment, trigger level 2 |
| 3 | Export 100 Folder Level emails with: 10 x 100 KB attachment, trigger level 2 |
| 4 | Export 100 Folder Level emails with: 10 x 1 MB attachment, trigger level 2 |
| 5 | WAN (like Scenario 1, 50 Mbps with 50 ms RT latency) 1 x 100 KB attachment, trigger level 2 |

| Scenario | Description |
|---|---|
| 6 | WAN (like Scenario 2, 50 Mbps with 50 ms RT latency) 1 x 1 MB attachment, trigger level 2 |
| 7 | WAN (like Scenario 3, 50 Mbps with 50 ms RT latency) 10 x 100 KB attachment, trigger level 2 |
| 8 | WAN (like Scenario 4, 50 Mbps with 50 ms RT latency) 10 x 1 MB attachment, trigger level 2 |

The above scenarios were tested using:

- 100 emails per batch where an email is a folder node (level 2).
- Email attachments are child document nodes located below each email node.
- Each email folder may have one attachment or 10 attachments (meaning 100 or 1000 attachments per batch). The worst case batch had 100 folders with 10 documents in each, which gives 1000 attachments (1 MB each), or 1 GB.
- The mail server is a Windows Server 2003 image with the SMTP role service.
- All tests were run at trigger level 2.

The data export scenarios were used to test how the export throughput is affected by the output file format (XML or CSV) and by the value mapping selection (UIMdata or MDF):

**Table 36. Data Export scenarios**

| Scenario | Description |
|---|---|
| 1 | UIM to XML 10 values, trigger level 1 |
| 2 | UIM to XML 106 values, trigger level 1 |
| 3 | UIM to XML 506 values, trigger level 1 |
| 4 | MDF to XML 10 values, trigger level 1 |
| 5 | MDF to XML 106 values, trigger level 1 |
| 6 | MDF to XML 506 values, trigger level 1 |
| 7 | MDF to CSV 10 values, trigger level 1 |
| 8 | MDF to CSV 106 values, trigger level 1 |
| 9 | MDF to CSV 506 values, trigger level 1 |
| 10 | Repeat of Scenario 1, trigger level 7 |
| 11 | Repeat of Scenario 5, trigger level 7 |
| 12 | Repeat of Scenario 9, trigger level 7 |

The above scenarios were tested with the following batch structure:

- 100 documents per batch.
- The number of pages per document varied depending on the data set size, but typically 3 to 5 pages.

## Benchmark Results

Refer to *Test Environment Used for Testing of Classification and* Extraction Modules

System Properties for Capture Client Modules and Administrator

| Item | Required |
|---|---|
| Hardware | Lenovo C30 Workstation<br>• Intel Xeon E5-2609 v2 @ 2.5 GHz (quad core)<br>• 4 virtual CPUs allocated to VM<br>• 4 GB RAM allocated to VM<br>• 1 Gbit Network interface card |
| Operating System | vSphere ESXi 5.5 as virtual machine host<br>VM image built with Windows Server 2012 R2 |

Explanation of Columns in Client Module Benchmark Results for explanation of the columns in the benchmark results tables.

## File Export Scenarios

**Table 37. One Instance of SE Running on a Single-Core Machine — File Export**

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| **Trigger Level 1** | | | | | |
| Export Doc Level stage files, file size = 1 MB, 100 files per batch, trigger level 1 | 77,362 | 18% | 36 | 22 / 22 | 6 / 6 |
| Export Doc Level stage files, file size = 10 MB, 100 files per batch, trigger level 1 | 13,864 | 63% | 83 | 40 / 41 | 3 / 117 |
| Export Doc Level stage files, file size = 100 MB, 100 files per batch, trigger level 1 | 579 | 85% | 271 | 17 / 15 | 4 / 4 |
| **Trigger Level 7** | | | | | |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| Export Doc Level stage files, file size = 1 MB, 100 files per batch, trigger level 7 | 91,139 | 37% | 134 | 26 / 26 | 0 / 0 |
| Export Doc Level stage files, file size = 10 MB, 100 files per batch, trigger level 7 | 8,920 | 62% | 181 | 29 / 30 | 0 / 0 |
| Export Doc Level stage files, file size = 100 MB, 100 files per batch, trigger level 7 | 580 | 85% | 284.32 | 17 / 17 | 0 / 0 |

**Table 38. Multiple Instances of SE Running on a Multi-Core Machine — File Export**

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) |
|---|---|---|---|---|---|
| **Export Doc Level stage files, file size = 1 MB, 100 files per batch, trigger level 1** | | | | | |
| 1 Instance, 1 CPU core | 77,362 | 1.00 | 18 | 36 | 22 / 22 |
| 2 Instances, 2 CPU cores | 100,840 | 1.30 | 11 | 98 | 26 / 26 |
| 4 Instances, 4 CPU cores | 135,593 | 1.75 | 9 | 143 | 37 / 37 |
| **Export Doc Level stage files, file size = 10 MB, 100 files per batch, trigger level 1** | | | | | |
| 1 Instance, 1 CPU core | 13,864 | 1.00 | 63 | 83 | 40 / 41 |
| 2 Instances, 2 CPU cores | 18,265 | 1.37 | 55 | 92 | 52 / 53 |
| 4 Instances, 4 CPU cores | 23,529 | 1.69 | | | |
| **Export Doc Level stage files, file size = 100 MB, 100 files per batch, trigger level 1** | | | | | |

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) |
|---|---|---|---|---|---|
| 1 Instance, 1 CPU core | 579 | 1.00 | 85 | 271 | 17 / 15 |
| 2 Instances, 2 CPU cores | 755 | 1.30 | 87 | 278 | 22 / 22 |
| 4 Instances, 4 CPU cores | 920 | 1.58 | 95 | 591 | 27 / 35 |
| **Export Doc Level stage files, file size = 1 MB, 100 files per batch, trigger level 7** | | | | | |
| 1 Instance, 1 CPU core | 91,139 | 1.00 | 37 | 159 | 26 / 26 |
| 2 Instances, 2 CPU cores | 112,500 | 1.23 | 25 | 192 | 32 / 33 |
| 4 Instances, 4 CPU cores | 129,496 | 1.42 | 26 | 498 | 37 / 37 |
| **Export Doc Level stage files, file size = 10 MB, 100 files per batch, trigger level 7** | | | | | |
| 1 Instance, 1 CPU core | 8,920 | 1.00 | 62 | 134 | 29 / 30 |
| 2 Instances, 2 CPU cores | 16,636 | 1.86 | 45 | 224 | 45 / 45 |
| 4 Instances, 4 CPU cores | 20,033 | 2.24 | 50 | 430 | 58 / 59 |
| **Export Doc Level stage files, file size = 100 MB, 100 files per batch, trigger level 7** | | | | | |
| 1 Instance, 1 CPU core | 580 | 1.00 | 85 | 284 | 17 / 17 |
| 2 Instances, 2 CPU cores | 716 | 1.23 | 88 | 542 | 21 / 21 |
| 4 Instances, 4 CPU cores | 888 | 1.53 | 91 | 1084 | 26 / 26 |

## Email Export Scenarios

**Table 39. One Instance of SE Running on a Single-Core Machine — Email Export**

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| **Trigger Level 2** | | | | | |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| Export 100 Folder Level emails with: 1x100 KB attachment, trigger level 2 | 124,352 | 21.61 | 36.12 | 4 / 5 | 0.00 / 1 |
| Export 100 Folder Level emails with: 1x1 MB attachment, trigger level 2 | 23,576 | 27.34 | 171.62 | 7 / 9 | 0.00 / 0.00 |
| Export 100 Folder Level emails with: 10x100 KB attachment, trigger level 2 | 17,634 | 22.76 | 70.72 | 5 / 7 | 0.00 / 0.00 |
| Export 100 Folder Level emails with: 10x1 MB attachment, trigger level 2 | 3,169 | 35.31 | 84.54 | 9 / 13 | 0.00 / 0.00 |
| **Trigger Level 2, WAN 50 MS** | | | | | |
| WAN (like S9, 50 Mbps with 50 ms RT latency) 1x100 KB attachment, trigger level 2 | 3,842 | 0.95 | 35.88 | 123 / 158 | 2.42 / 5.16 |
| WAN (like S9, 50 Mbps with 100 ms RT latency) 1x1 MB attachment, trigger level 2 | 487 | 2.70 | 40.69 | 152 / 202 | 3.10 / 7.36 |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| WAN (like S9, 50 Mbps with 150 ms RT latency)<br>10 x 100 KB attachment, trigger level 2 | 487 | 0.79 | 41.13 | 150 / 198 | 2.74 / 4.82 |
| WAN (like S9, 50 Mbps with 150 ms RT latency)<br>10 x 1 MB attachment, trigger level 2 | 50 | 0.91 | 66.03 | 156 / 207 | 9.38 / 6.08 |

**Table 40. One Instance of SE Running on a Multi-Core Machine — Email Export**

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) |
|---|---|---|---|---|---|
| **Export 100 Folder Level emails with: 1 x 100 KB attachment, trigger level 2** | | | | | |
| 1 Instance, 1 CPU core | 124,352 | 1.00 | 21.61 | 36.12 | 4 / 5 |
| 2 Instances, 2 CPU cores | 155,340 | 1.24 | 12 | 73.25 | 4 / 6 |
| 4 Instances, 4 CPU cores | 189,225 | 1.52 | 8 | 146 | 5.21 / 6.84 |
| **Export 100 Folder Level emails with: 1 x 1 MB attachment, trigger level 2** | | | | | |
| 1 Instance, 1 CPU core | 23,576 | 1.00 | 27.34 | 39.83 | 7 / 9 |
| 2 Instances, 2 CPU cores | 27,057 | 1.14 | 15 | 87.28 | 8 / 11 |
| 4 Instances, 4 CPU cores | 33,012 | 1.40 | 11.34 | 174.56 | 9.78 / 13.24 |
| **Export 100 Folder Level emails with: 10 x 100 KB attachment, trigger level 2** | | | | | |
| 1 Instance, 1 CPU core | 17,634 | 1.00 | 22.76 | 70.72 | 5 / 7 |
| 2 Instances, 2 CPU cores | 25,105 | 1.42 | 17 | 113.62 | 7 / 10 |
| 4 Instances, 4 CPU cores | 30,000 | 1.70 | 11.31 | 227.24 | 8.86 / 11.96 |

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (MB) |
|---|---|---|---|---|---|
| **Export 100 Folder Level emails with: 10 x 1 MB attachment, trigger level 2** | | | | | |
| 1 Instance, 1 CPU core | 3,169 | 1.00 | 35.31 | 84.54 | 9 / 13 |
| 2 Instances, 2 CPU cores | 4,103 | 1.29 | 22 | 163.45 | 12 / 16 |
| 4 Instances, 4 CPU cores | 8,075 | 2.54 | 16.78 | 326.89 | 15.36 / 20.84 |

## Data Export Scenarios

**Table 41. One Instance of SE Running on a Single-Core Machine — Data Export**

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| **Trigger Level 1** | | | | | |
| UIM to XML 10 values, trigger level 1 | 87,379 | 30.90 | 42.28 | 1,839 / 102 | 15 / 6 |
| UIM to XML 106 values, trigger level 1 | 104,046 | 47.65 | 52.53 | 2,631 / 292 | 2 / 7 |
| UIM to XML 506 values, trigger level 1 | 37,736 | 59.63 | 54.77 | 4,081 / 437 | 2 / 7 |
| MDF to XML 10 values, trigger level 1 | 179,104 | 40.94 | 56.77 | 1,447 / 328 | 0 / 8 |
| MDF to XML 106 values, trigger level 1 | 53,651 | 37.66 | 53.13 | 348 / 343 | 0 / 6 |
| MDF to XML 506 values, trigger level 1 | 13,534 | 38.24 | 55.07 | 304 / 368 | 1 / 4 |
| MDF to CSV 10 values, trigger level 1 | 114,650 | 46.33 | 55.47 | 2,233 / 213 | 0 / 8 |
| MDF to CSV 106 values, trigger level 1 | 46,213 | 44.7 | 53.50 | 1,512 / 239 | 0 / 5 |

| Scenario | Throughput (Pages per Hour) | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) | Disk Usage Read / Write per Second (KB) |
|---|---|---|---|---|---|
| MDF to CSV 506 values, trigger level 1 | 10,883 | 44.48 | 58.26 | 1,419 / 254 | 1 / 4 |
| **Trigger Level 7** | | | | | |
| UIM to XML 10 values, trigger level 7 | 166,667 | 50.79 | 59.45 | 3,221/166 | |
| MDF to XML 106 values, trigger level 7 | 61,750 | 40.75 | 57.35 | 942 / 318 | 0 / 4 |
| MDF to CSV 506 values, trigger level 7 | 11,400 | 45.04 | 83.74 | 1433 / 261 | 8 / 7 |

**Table 42. One Instance of SE Running on a Multi-Core Machine — Data Export**

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| **UIM to XML 10 values, trigger level 1** | | | | | |
| 1 Instance, 1 CPU core | 87,379 | 1.00 | 30.90 | 42.28 | 1,839 / 102 |
| 2 Instances, 2 CPU cores | 111,455 | 1.27 | 22.01 | 83.93 | 2,801 / 167 |
| 4 Instances, 4 CPU cores | 162,162 | 1.85 | 16.88 | 151.01 | 3,206 / 324 |
| **UIM to XML 106 values, trigger level 1** | | | | | |
| 1 Instance, 1 CPU core | 104,046 | 1.00 | 56.2 | 52.53 | 2,631 / 292 |
| 2 Instances, 2 CPU cores | 159,292 | 1.53 | 32.3 | 88.22 | 628 / 661 |
| 4 Instances, 4 CPU cores | 270,677 | 2.6 | 21.85 | 177.32 | 4,224 / 560 |
| **UIM to XML 506 values, trigger level 1** | | | | | |
| 1 Instance, 1 CPU core | 37,736 | 1.00 | 59.63 | 52.53 | 2,631 / 292 |
| 2 Instances, 2 CPU cores | 50,847 | 1.34 | 37.13 | 94.93 | 5,308 / 572 |
| 4 Instances, 4 CPU cores | 66,667 | 1.76 | 26.53 | 172.64 | 7,413 / 749 |
| **MDF to XML 10 values, trigger level 1** | | | | | |

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| 1 Instance, 1 CPU core | 179,104 | 1.00 | 40.94 | 32.87 | 1,447 / 328 |
| 2 Instances, 2 CPU cores | 251,748 | 1.40 | 29.80 | 91.75 | 985 / 557 |
| 4 Instances, 4 CPU cores | 473,684 | 2.64 | 32.73 | 177.10 | 4,235 / 958 |
| MDF to XML 106 values, trigger level 1 | | | | | |
| 1 Instance, 1 CPU core | 53,651 | 1.00 | 26.52 | 32.87 | 348 / 343 |
| 2 Instances, 2 CPU cores | 93,023 | 1.74 | 29.18 | 89.77 | 581 / 580 |
| 4 Instances, 4 CPU cores | 174,757 | 3.25 | 34.03 | 173.1 | 1,188 / 1,022 |
| MDF to XML 506 values, trigger level 1 | | | | | |
| 1 Instance, 1 CPU core | 13,534 | 1.00 | 26.47 | 55.07 | 304 / 368 |
| 2 Instances, 2 CPU cores | 20,282 | 1.49 | 25.90 | 97.04 | 459 / 552 |
| 4 Instances, 4 CPU cores | 44,610 | 3.29 | 35.83 | 175.37 | 1,008 / 1,204 |
| MDF to CSV 10 values, trigger level 1 | | | | | |
| 1 Instance, 1 CPU core | 114,650 | 1.00 | 38.87 | 55.47 | 2,233 / 213 |
| 2 Instances, 2 CPU cores | 102,857 | N/A | 24.61 | 102.21 | 2,332 / 221 |
| 4 Instances, 4 CPU cores | 202,247 | 1.76 | 28.41 | 147.76 | 3,959 / 556 |
| MDF to CSV 106 values, trigger level 1 | | | | | |
| 1 Instance, 1 CPU core | 46,213 | 1.00 | 38.87 | 55.47 | 1,512 / 239 |
| 2 Instances, 2 CPU cores | 45,113 | N/A | 22.05 | 101.18 | 1425 / 329 |
| 4 Instances, 4 CPU cores | 117,264 | 2.53 | 27.87 | 177.83 | 3,853 / 557 |
| MDF to CSV 506 values, trigger level 1 | | | | | |
| 1 Instance, 1 CPU core | 10,883 | 1.00 | 44.48 | 58.26 | 1,419 / 254 |
| 2 Instances, 2 CPU cores | 20,067 | 1.84 | 38.09 | 106.75 | 2,583 / 463 |

| Scenario | Throughput (Pages per Hour) | Effective Units of Full Module Performance | CPU % Processor Time | Private Bytes (MB) | Network Data Received / Sent per Second (KB) |
|---|---|---|---|---|---|
| 4 Instances, 4 CPU cores | 25,550 | 2.34 | 28.73 | 199.98 | 3,381 / 609 |
| **UIM to XML 10 values, trigger level 7** | | | | | |
| 1 Instance, 1 CPU core | 166,667 | 1.00 | 50.79 | 59.45 | 3,221 / 166 |
| 2 Instances, 2 CPU cores | 318,584 | 1.91 | 41.55 | 114.70 | 2,878 / 573 |
| 4 Instances, 4 CPU cores | 400,00 | 2.39 | 29.16 | 211.83 | 5,468 / 443 |
| **MDF to XML 106 values, trigger level 7** | | | | | |
| 1 Instance, 1 CPU core | 61,750 | 1.00 | 40.75 | 57.35 | 942 / 318 |
| 2 Instances, 2 CPU cores | 129,964 | 2.10 | 39.75 | 125.99 | 1,945 / 656 |
| 4 Instances, 4 CPU cores | 232,258 | 3.76 | 40.42 | 226.69 | 5,133 / 934 |
| **MDF to CSV 506 values, trigger level 7** | | | | | |
| 1 Instance, 1 CPU core | 11,400 | 1.00 | 44.48 | 58.26 | 1,419 / 254 |
| 2 Instances, 2 CPU cores | 23,514 | 2.06 | 42.70 | 166.18 | 2,976 / 538 |
| 4 Instances, 4 CPU cores | 27,544 | 2.41 | 31.26 | 314.6 | 3,381 / 609 |

## Summary of Results

- Larger files export slower than small files. Throughput of file export can be increased by 20% when triggering at higher level (7 vs 1). However, the higher trigger level does not improve the throughput for large file size (100MB) as most of the file export time is spent to transfer files to the file share.
- When exporting files to a network share on the WAN, the throughput can be impacted. Though this scenario was not tested, looking at the amount of network traffic, we can assume that higher latency on the network will have an impact on the module's throughput (pages/hr).
- The attachment size and number of attachments in the task have an impact on the module's throughput (email/hr). If the exported attachment is only 100 KB in size, by increasing the number of attachments in the task by 10 times the module throughput (email/hr) gets at least seven times slower as compared to one attachment in the task.
- It is recommended to export emails to a local mail server rather than to a mail server over the WAN. The throughput of email export significantly drops with the RTT latency of 50 ms. With higher latency, Standard Export encounters errors.

- When exporting data, the number of data values being exporting has an impact on the module's throughput.
- Value mapping selection affects the data export throughput in certain scenarios. MDF value mapping showed much higher efficiency when exporting MDF values for every small number of values, while exporting large amount of table data was more efficient with UIMdata value mapping. Exporting UIMdata values puts less burden on the InputAccel Server as compared to MDF data values. Though, when comparing MDF export vs. UIMdata export, the CPU usage on the InputAccel Server was low in both cases: 5% average for MDF value export and less than 2% when exporting UIMdata.
- When exporting data to XML vs. CSV files, throughput was higher when exporting data values to XML files as compared to CSV.
- As Standard Export is not a CPU-intensive module, it is typically possible to deploy more instances of Standard Export than CPU cores. However, throughput may not double when the second instance is deployed.

### Critical Factors Affecting Standard Export Performance

- Standard Export requires high bandwidth and low latency to achieve optimum throughput.
- Throughput drops when exporting large files at the document level.
- For email export, large files and a high number of attachments slows throughput. Fast network bandwidth and low RTT latency is required for optimum throughput.
- Exporting MDF data values as small amounts of data (e.g. 10 fields) shows higher throughput than exporting the same set of fields as UIMdata values. The drawback to using MDF values is that flatten data values lose their original data type. Flatten data is output as strings if it is not defined in the MDF values of a step which data values were flatten.
- Triggering at a higher level 7 increases the module throughput when exporting large or small document data values. When exporting large size or amount of data values, triggering at level 7 didn't produce any significant increase of the module throughput.

### Non-Critical Factors

- **Disk usage:** Disk usage is an insignificant factor. 50 GB free disk space is recommended, although the typical usage is less than 1 GB.

## Standard Import

Standard Import module imports image and non-image files from file directories as well as emails and attachments from email servers. Imported files can be stored in the batch at document or page level. Rules can be applied to filter only certain types of files, emails, or attachments to be imported. Standard Import is designed to replace the Multi-Directory Watch and Email Import modules.

The tests performed for Standard Import fall into two usage categories: File import and Email import.

## Test Scenarios for File Import

The following file import scenarios were used to test various impacts of size of the files imported, quantity of files imported into each batch, filtering of files based on extension, importing supplemental data values through data file, and importing files over a WAN connection.

**Table 43. File Import Scenarios**

| Scenario | Description |
|---|---|
| **Impact of # of files per batch (medium size at document level)** | |
| **Scenario 1**:<br>Small batch<br>(1 document per batch) | Files per batch: 1<br>Size of each file: 10 MB (10 MB file data total)<br>Level of file: Document<br>Compare with : Scenario 2 (more files per batch)<br>Scenario 3 (more files per batch)<br>Scenario 6 (smaller file size) |
| **Scenario 2**:<br>Medium batch<br>(10 documents per batch) | Files per batch: 10<br>Size of each file: 10 MB (100 MB file data total)<br>Level of file: Document<br>Compare with: Scenario 1 (fewer files per batch)<br>Scenario 3 (more files per batch)<br>Scenario 4 (larger file size)<br>Scenario 5 (file filtering)<br>Scenario 7 (smaller file size)<br>Scenario 10 (50 ms WAN latency)<br>Scenario 11 (100 ms WAN latency) |
| **Scenario 3**:<br>Large batch<br>(100 documents per batch) | Files per batch: 100<br>Size of each file: 10 MB (nearly 1 GB file data total)<br>Level of file: Document<br>Compare with: Scenario 1 (fewer files per batch)<br>Scenario 2 (fewer files per batch)<br>Scenario 8 (smaller file size) |

| Impact of large files | |
|---|---|
| **Scenario 4**:<br>Large file size | Files per batch: 10<br>Size of each file: 100 MB (nearly 1 GB file data total)<br>Level of file: Document<br>Compare with: Scenario 2 (smaller file size) |
| **Impact of file filtering** | |
| **Scenario 5**:<br>File filtering | Files per batch: 10<br>Size of each file: 10 MB (PDF) + 1 KB (TXT)<br>Level of file: Document<br>Special rule: Only import PDF files to batch (skip other files)<br>TXT files were removed but not imported into batch<br>Compare with: Scenario 2 (no file filtering) |
| **Impact of # of files per batch (small size at page level)** | |
| **Scenario 6**:<br>Very small batch<br>(1 page per batch) | Files per batch: 1<br>Size of each file: 63 KB (average) (63 KB file data total)<br>Level of file: Page<br>Compare with: Scenario 1 (larger file size) |
| **Scenario 7**:<br>Small batch<br>(10 pages per batch) | Files per batch: 10<br>Size of each file: 63 KB (average) (630 KB file data total)<br>Level of file: Page<br>Compare with: Scenario 2 (larger file size) |
| **Scenario 8**:<br>Medium batch<br>(100 pages per batch) | Files per batch: 100<br>Size of each file: 63 KB (average) (6.2 MB file data total)<br>Level of file: Page<br>Compare with: Scenario 3 (larger file size)<br>Scenario 9 (with data file) |
| **Impact of data file** | |
| **Scenario 9**:<br>Data file | Files per batch: 100<br>Size of each file: 63 KB (average) TIFF + TXT (10 data values)<br>Level of file: Page<br>Compare with: Scenario 8 (without data file) |

| Impact of WAN | | |
|---|---|---|
| **Scenario 10**:<br>WAN 50 ms | Files per batch:<br>Size of each file:<br>Level of file:<br>WAN Latency:<br>Compare with:<br>NOTE 1:<br>NOTE 2: | 10<br>10 MB<br>Document<br>50 ms<br>Scenario 2 (no WAN latency)<br>Both input and output folders were over WAN<br>Latency is round trip, as reported by "ping" |
| **Scenario 11**:<br>WAN 100 ms | Files per batch:<br>Size of each file:<br>Level of file:<br>WAN Latency:<br>Compare with:<br>NOTE 1:<br>NOTE 2: | 10<br>10 MB<br>Document<br>100 ms<br>Scenario 2 (no WAN latency)<br>Both input and output folders were over WAN<br>Latency is round trip, as reported by "ping" |

## Benchmark Results for File Import

The scenarios described previously were performed to test various sizes of the imported files, quantity of files imported per batch, filtering of files based on extension, importing data values through data file, and files over WAN connection. Refer back to the scenario descriptions as needed.

The results that follow are given in terms of "files per hour" indicating how many files from the import folder could be ingested into batches in one hour. Additionally, the equivalent "MB/hour" rate (based on the total size of all files ingested) is given to better understand the total quantity of file data that has been imported.

NOTE: Only one instance of Standard Import running on a Single-Core Machine was tested

**Table 44. File Import Test Results**

| Scenarios | Throughput in files/hour (docs or pages) and MB/hour | CPU % Processor Time (total) | Network Data MB Received per second | Network Data MB Sent per second | Comments |
|---|---|---|---|---|---|
| **Impact of # files per batch (medium size at document level)** | | | | | |
| **Scenario 1**: Small batch (1 document per batch) | 9,474 docs/hour<br>94,740 MB/hour | 32 | 26 | 26 | Importing very few files per batch is less efficient and hurts file ingestion rate, but only slightly, as seen in Scenarios 2 and 3. |
| **Scenario 2**: Medium batch (10 documents per batch) | 10,651 docs/hour<br>106,510 MB/hour | 34 | 31 | 30 | See previous comment. |
| **Scenario 3**: | 10,876 docs/hour | 35 | 31 | 30 | See previous comment. |

| Scenarios | Throughput in files/hour (docs or pages) and MB/hour | CPU % Processor Time (total) | Network Data MB Received per second | Network Data MB Sent per second | Comments |
|---|---|---|---|---|---|
| Large batch (100 documents per batch) | 108,760 MB/hour | | | | |
| **Impact of large files** | | | | | |
| **Scenario 4**: Large file size | 1,434 docs/hour<br><br>143,400 MB/hour | 38 | 35 | 34 | Larger files naturally take longer to import, but they import more efficiently than small files in terms of net bytes per hour. Compare with Scenario 2. |
| **Impact of file filtering** | | | | | |
| **Scenario 5**: File filtering | 10,843 docs/hour<br><br>108,430 MB/hour | 36 | 32 | 31 | Filtering out unwanted small files did not have any impact. Result was essentially the same as Scenario 2. |

| Scenarios | Throughput in files/hour (docs or pages) and MB/hour | CPU % Processor Time (total) | Network Data MB Received per second | Network Data MB Sent per second | Comments |
|---|---|---|---|---|---|
| **Impact of # files per batch (small size at page level)** | | | | | |
| **Scenario 6**: Very small batch (1 page per batch) | 54,381 pages/hour<br>3,346 MB/hour | 14 | 1 | 1 | Small files are more sensitive to efficiencies of batch size. Importing larger quantities of small files per batch improves performance as seen in Scenarios 7 and 8. |
| **Scenario 7**: Small batch (10 pages per batch) | 174,757 pages/hour<br>10,752 MB/hour | 14 | 3 | 3 | See previous comment. |
| **Scenario 8**: Medium batch (100 pages per batch) | 262,774 pages/hour<br>16,167 MB/hour | 14 | 4 | 4 | See previous comment. |
| **Impact of data file** | | | | | |
| **Scenario 9**: Data file | 209,302 pages/hour<br>12,877 MB/hour | 21 | 3 | 3 | Importing data file (10 values extracted per page-level image imported) will slow performance. Compare these rates with Scenario 8. |
| **Impact of WAN** | | | | | |
| **Scenario 10**: WAN 50 ms | 1,689 docs/hour<br>16,890 MB/hour | 7 | 5 | 5 | WAN latency of 50 ms (net round trip latency) slowed the import to be just 16% of the ideal rate with no latency, as compared with Scenario 2. |
| **Scenario 11**: WAN 100 ms | 1,033 docs/hour<br>10,330 MB/hour | 5 | 3 | 3 | WAN latency of 100 ms (net round trip latency) slowed the import to be just 10% of the ideal rate with no latency, as compared with Scenario 2. |

## Conclusions on File Import

### Effect of Batch Size (number of files per batch):

Very small batches (small number of files) result in additional overhead for creating more batches more frequently on the InputAccel Server. This additional overhead reduces the efficiency or rate at which files can be imported.

Very small batches should be avoided if possible. Keep in mind, however, that very large batches can be less efficient for downstream modules to process, so a good balance must be found.

**Effect of File Size (size of file imported):**

A small number of large files is faster to import than a large number of small files. It is more efficient to import multipage files at document level than to import single page images at page level.

Multipage files do, however, need to be processed subsequently with Image Converter to break them into single page images, but that procedure can be easily scaled up with additional Image Converter instances. If file import rate is a concern, importing data as multipage files can help.

**Effect of File Filtering:**

Filtering of files did not have any significant impact on performance.

**Effect of Data File:**

Importing supplemental metadata through a data file (one data file per imported file) takes additional time and can impact performance.

**Effect of WAN:**

Importing over a WAN can be significantly slower. Possibilities to mitigate this impact include

- Replicate the import folder to a location on the same LAN
- Locate at least the "output" folder on a same LAN, so that only the "import" folder is over WAN connection
- Reduce the number of files "seen" in the import folder during any given directory poll by increasing the poll interval. When large numbers files are found in a folder, it takes more time to query and analyze the timestamps prior to importing the files.
- Import larger files if possible, such as multipage instead of single page
- If possible, partition the contents of the import folder into multiple folders and use separate instances of Standard Import importing at the same time from different folders.

NOTE: never configure multiple instances of Standard Import to import from the same folder at the same time.

## Test Scenarios for Email Import

The following email import scenarios were used to test various impacts of attachment size, attachment quantity, filtering of emails based on attachment type, number of emails per batch, use of different email protocols, email forwarding, and email retrieval and forwarding over WAN conditions.

Unless otherwise stated

- For every batch, 10 emails were imported
- IMAP protocol was used for retrieving the emails from a Microsoft Exchange 2013 server
- Network bandwidth was 50 Mbps
- Network latency was 0 ms

Note: The default client throttling policies in Microsoft Exchange Server were overridden (disabled) to allow consistent performance measurements to be made

**Table 45. Email Import Scenarios**

| Scenario | Description | |
|---|---|---|
| **Impact of Size and Quantity of Attachments** | | |
| **Scenario 1**:<br>Single attachment<br>(medium size) | Attachments per email:<br>Size of attachment:<br>Compare with : | 1<br>1 MB<br>Scenario 2 (multiple attachments)<br>Scenario 3 (larger file attachment)<br>Scenario 4 (email filtering)<br>Scenario 5 (1 email per batch) |
| **Scenario 2**:<br>Multiple attachments<br>(medium size) | Attachments per email:<br>Size of each attachment:<br>Compare with : | 5<br>1 MB<br>Scenario 1 (single attachment)<br>Scenario 6 (email forwarding)<br>Scenario 7 (50 ms WAN latency)<br>Scenario 8 (100 ms WAN latency)<br>Scenario 11 (POP3 Protocol)<br>Scenario 12 (EWS Protocol) |
| **Scenario 3**:<br>Single attachment<br>(large size) | Attachments per email:<br>Size of attachment:<br>Compare with : | 1<br>10 MB<br>Scenario 1 (smaller file attachment) |

| Impact of Email Filtering | | |
|---|---|---|
| **Scenario 4**:<br>Email Filtering | Attachments per email:<br>Size of imported attachment:<br>Email Filter Rule:<br>Attachment Types:<br><br><br><br><br>Compare with : | 1<br>1 MB<br>Import email only if PDF attachment exists<br>50% of emails had 1 MB PDF<br>50% of emails had 10 KB TXT<br>Only emails with PDF were imported<br>Other emails with TXT were skipped entirely<br>Scenario 1 (no email filtering) |
| **Impact of Emails per Batch** | | |
| **Scenario 5**:<br>Emails per Batch | Emails per Batch:<br>Attachments per email:<br>Size of attachment:<br>Compare with : | 1<br>1<br>1 MB<br>Scenario 1 (10 emails per batch) |
| **Impact of Email Forwarding** | | |
| **Scenario 6**:<br>Email Forwarding | Attachments per email:<br>Size of each attachment:<br>Protocol (forwarding):<br>Compare with : | 5<br>1 MB<br>SMTP<br>Scenario 2 (no email forwarding)<br>Scenario 9 (50 ms WAN latency to SMTP)<br>Scenario 10 (100 ms WAN latency to SMTP) |
| **Impact of WAN Latency on Retrieval from Email Server** | | |
| **Scenario 7**:<br>Retrieving over WAN (50 ms) | Attachments per email:<br>Size of each attachment:<br>WAN latency:<br>Compare with :<br><br>NOTE: Latency is round trip, as reported by “ping” | 5<br>1 MB<br>50 ms<br>Scenario 2 (0 ms WAN latency)<br>Scenario 8 (100 ms WAN latency) |
| **Scenario 8**:<br>Retrieving over WAN (100 ms) | Attachments per email:<br>Size of each attachment:<br>WAN latency:<br>Compare with :<br><br>NOTE: Latency is round trip, as reported by “ping” | 5<br>1 MB<br>100 ms<br>Scenario 2 (0 ms WAN latency)<br>Scenario 7 (50 ms WAN latency) |

| Impact of WAN Latency on Forwarding to SMTP Server (only) | |
|---|---|
| **Scenario 9**:<br>Forwarding over WAN (50 ms) | Attachments per email: 5<br>Size of each attachment: 1 MB<br>WAN latency: 50 ms<br>Protocol (forwarding): SMTP<br>Compare with : Scenario 6 (0 ms WAN latency to SMTP)<br>Scenario 10 (100 ms WAN latency to SMTP)<br>NOTE 1: Latency is round trip, as reported by "ping"<br>NOTE 2: Retrieval of emails was done with no latency. The only latency was on the connection to the forwarding SMTP server. |
| **Scenario 10**:<br>Forwarding over WAN (100 ms) | Attachments per email: 5<br>Size of each attachment: 1 MB<br>WAN latency: 100 ms<br>Protocol (forwarding): SMTP<br>Compare with : Scenario 6 (0 ms WAN latency to SMTP)<br>Scenario 9 (50 ms WAN latency to SMTP)<br>NOTE 1: Latency is round trip, as reported by "ping"<br>NOTE 2: Retrieval of emails was done with no latency. The only latency was on the connection to the forwarding SMTP server. |
| **Impact of Email Retrieval Protocol** | |
| **Scenario 11**:<br>POP3 Protocol | Attachments per email: 5<br>Size of each attachment: 1 MB<br>Protocol (receiving): POP3<br>Compare with : Scenario 2 (IMAP protocol)<br>Scenario 12 (EWS protocol) |
| **Scenario 12**:<br>EWS Protocol | Attachments per email: 5<br>Size of each attachment: 1 MB<br>Protocol (receiving): EWS (Exchange Web Services)<br>Compare with : Scenario 2 (IMAP protocol)<br>Scenario 11 (POP3 protocol) |

## Benchmark Results for Email Import

The scenarios described previously were performed to test various impacts of attachment size, attachment quantity, filtering of emails based on attachment type, number of emails per batch, use of different email protocols, email forwarding, and email retrieval and forwarding over WAN conditions. Refer back to the scenario descriptions as needed.

The results that follow are given in terms of "emails per hour" indicating how many emails could be retrieved from the email server in one hour. Additionally, the equivalent "Attachment MB/hour" rate (based on the total size of all the file attachments in the emails) is given to better understand the total quantity of file data that has been imported.

NOTE: Only one instance of Standard Import running on a Single-Core Machine was tested

**Table 46. Email Import Test Results**

| Scenarios | Throughput in Emails/hour and Attachment MB/ hour | CPU % Processor Time (total) | Network Data MB Received per second | Network Data MB Sent per second | Comments |
|---|---|---|---|---|---|
| **Impact of Size and Quantity of Attachments** | | | | | |
| **Scenario 1**: Single attachment (medium size) | 13,284 emails/hour<br><br>13,284 att MB/hour | 16 | 10 | 8 | Simple emails with 1 attachment can be processed very quickly (more than 3 emails per second). |
| **Scenario 2**: Multiple attachments (medium size) | 3,258 emails/hour<br><br>16,290 att MB/hour | 19 | 12 | 10 | Emails with multiple attachments are processed more efficiently than emails with single attachments. Total rate of attachment file MB imported is greater in this scenario. |
| **Scenario 3**: Single attachment (large size) | 1,709 emails/hour<br><br>17,090 att MB/hour<br>(1,709 att/hour) | 17 | 13 | 10 | Larger attachments naturally take longer to import, but they import more efficiently as seen here in the highest attachment MB rate. |

| Scenarios | Throughput in Emails/hour and Attachment MB/ hour | CPU % Processor Time (total) | Network Data MB Received per second | Network Data MB Sent per second | Comments |
|---|---|---|---|---|---|
| **Impact of Email Filtering** | | | | | |
| **Scenario 4**: Email Filtering | 12,329 emails/hour<br><br>12,329 att MB/hour | 17 | 8 | 6 | Filtering and thus skipping some emails adds only slightly to the total processing time. The rate here is only slightly lower than Scenario 1 which did not filter. |
| **Impact of Emails per Batch** | | | | | |
| **Scenario 5**: Emails per Batch | 10,619 emails/hour<br><br>10,619 att MB/hour | 30 | 8 | 7 | Importing fewer emails per batch (such as 1 email per batch) is less efficient and therefore slower. Compare the rate of emails/hour here to Scenario 1 where 10 emails per batch were imported. |
| **Impact of Email Forwarding** | | | | | |
| **Scenario 6**: Email Forwarding | 2,057 emails/hour<br><br>10,285 att MB/hour | 27 | 8 | 10 | Forwarding emails to an SMTP server on the same LAN can reduce overall performance to be roughly 2/3 of the rate seen when no forwarding was used, as in Scenario 2. |
| **Impact of WAN Latency on Retrieval from Email Server** | | | | | |
| **Scenario 7**: Retrieving over WAN (50 ms) | 1,234 emails/hour<br><br>6,170 att MB/hour | 6 | 5 | 4 | Importing from an email server with 50 ms latency can reduce the rate by more than half of what a LAN connection would produce, as in Scenario 2. |
| **Scenario 8**: Retrieving over WAN (100 ms) | 1,088 emails/hour<br><br>5,440 att MB/hour | 6 | 4 | 3 | Importing from an email server with 100 ms latency can reduce the rate to be about 1/3 of what a LAN connection would produce, as in Scenario 2. |

| Scenarios | Throughput in Emails/hour and Attachment MB/ hour | CPU % Processor Time (total) | Network Data MB Received per second | Network Data MB Sent per second | Comments |
|---|---|---|---|---|---|
| **Impact of WAN Latency on Forwarding to SMTP Server** | | | | | |
| **Scenario 9**: Forwarding over WAN (50 ms) | 438 emails/hour<br><br>2,190 att MB/hour | 7 | 2 | 2 | Forwarding emails to a server with 50 ms latency can reduce the rate to about 1/5 of what a LAN connection would produce, as in Scenario 6. |
| **Scenario 10**: Forwarding over WAN (100 ms) | 241 emails/hour<br><br>1,205 att MB/hour | 2 | 1 | 1 | Forwarding emails to a server with 100 ms latency can reduce the rate to about 1/10 of what a LAN connection would produce, as in Scenario 6. |
| **Impact of Email Retrieval Protocol** | | | | | |
| **Scenario 11**: POP3 Protocol | 3,349 emails/hour<br><br>16,745 att MB/hour | 26 | 12 | 10 | Use of POP3 protocol yielded roughly the same performance rate as IMAP. Compare this rate with Scenario 2. |
| **Scenario 12**: EWS Protocol | 2,344 emails/hour<br><br>11,720 att MB/hour | 21 | 10 | 7 | Use of the Exchange Web Services protocol reduced throughput to about 70% of what IMAP and POP3 achieved, as in Scenarios 2 and 11. |

## Conclusions on Email Import

### Effect of Attachment Size and Quantity:

Smaller attachments and more numerous attachments are less efficient to ingest than larger attachments in terms of net MB imported per hour. However, the difference is not drastic, and there is often little which can be done to control how emails are generated and submitted. If there are performance issues, one thing to consider would be asking for attachments to be zipped prior to sending.

### Effect of Email Filtering:

Filtering of emails resulted in only a small decrease in net throughput. It is not a significant factor for concern, although if skipped emails are forwarded to another SMTP server, there will be some additional performance penalty to consider.

### Effect of Batch Size (number of emails per batch):

Very small batches (small number of emails) result in additional overhead for creating more batches more frequently on the InputAccel Server. This additional overhead reduces the efficiency or rate at which emails can be imported. Very small batches should be avoided if possible. Keep in mind, however, that very large batches can be less efficient for downstream modules to process, so a good balance must be found.

**Effect of Email Forwarding:**

Forwarding of emails to an SMTP server can cut performance by 1/3. The speed at which the SMTP server can receive these emails will impact the rate at which emails can be imported. When deploying Standard Import, take into consideration the speed of both the incoming and outgoing email servers.

**Effect of WAN:**

Importing or forwarding over a WAN can be significantly slower. Possibilities to mitigate this impact include

- Co-locate the email server(s) in the same LAN as the Standard Import module
- Use an SMTP relay to allow forwarding to a local SMTP server which in turn forwards to a more distant server in the background
- Have the remote incoming email server forward emails to a local email server on the same LAN
- Partition incoming emails such that they come in to different mailboxes which can be independently processed by multiple Standard Import profiles running in parallel
- Use WAN accelerator/optimizer equipment

## Sizing Recommendations

Standard Import is primarily bound by disk and network performance, rather than CPU. From these test results, a single instance of Standard Import processing one profile consumed about 1/3 or less of a single CPU. If multiple instances of Standard Import will be run, a general recommendation would be to run no more than 2 instances per available CPU.

Predicting the performance of Standard Import with any given email server is difficult due to the unpredictability of the email server performance. If the email server is heavily loaded or intentionally throttling client access, the rate of email ingestion could be significantly reduced.

# Chapter 5 Captiva Web Client and REST Services

## Introduction

Captiva Capture Web Client is an easy-to-use, Web-based capture application that you can run in your browser at branch offices and other remote locations. It utilizes the Captiva REST Services to create batches on the InputAccel Server and to perform certain types of module processing on remote module servers.

Captiva REST Services are a set of RESTful web service interfaces that custom client applications can use to call the services of the InputAccel Server or Module Server. The Captiva Capture Web Client is one example of a Captiva REST Services client application.

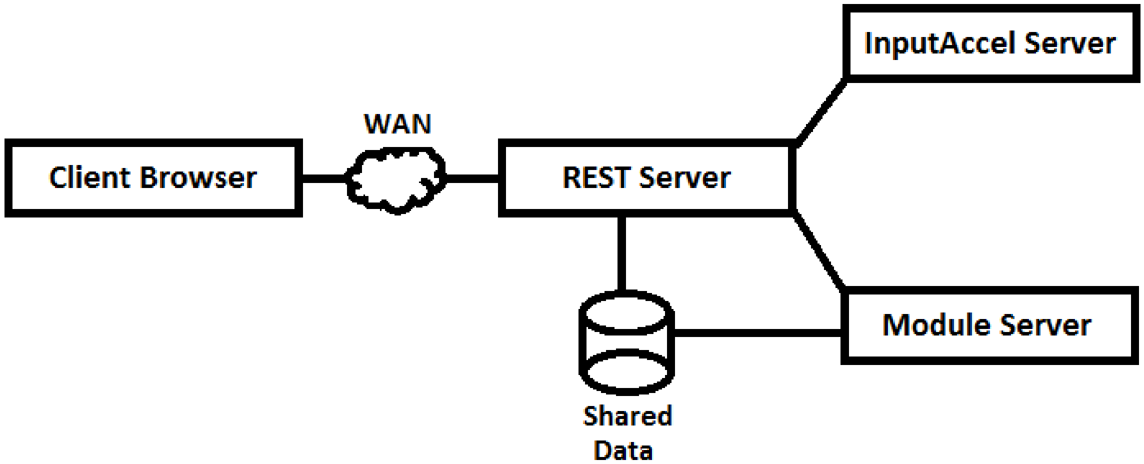
## Test Environment and Methodology

This section describes the testing methodology used to test and record benchmark results for Captiva Capture Web Client.

### Test Environment

The test environment consisted of the following systems, each running as a virtual machine:

- InputAccel Server
- REST Server (hosting the Captiva Capture Web Client site)
- Module Server (hosting client modules exposed as REST services)
- User desktop using Google Chrome browser to access CWC application
- Shared data file system with configuration data for REST server and Module Server




In tests under WAN conditions, the latency was applied on the connection between the browser (scan client) and the REST server (web server).

## Method of Testing

Testing of the Captiva Capture Web Client was performed with a custom automation script to “drive” the application to scan and submit multiple batches in sequence for each of the scenarios described below. This automation also captured critical timestamps about when each page was scanned, when the submit button became clickable, and when the submit operation completed.

The tests were targeted to measure 3 overall usage aspects:

- Scanning time (real time rate of scanning and total time spent scanning)
- User wait time (if any) after scanning is complete, but before batch is ready for user action
- Total time spent submitting the entire batch to the web server

Network conditions tested:

- Bandwidth = 10 Mbps (common for an inexpensive broadband internet connection)
- Network latency (round trip or “ping” latency to the web server)
    - 0 ms
    - 50 ms
    - 150 ms

Unless otherwise stated, all scenarios used batches of

- Bitonal TIFF images, 300 DPI, average size 30 KB
- 200 images per batch
- Batches automatically split into 50 documents (4 pages each)

Scanning was performed with a simulated scanner using the “Demo Driver” which imports files from file system and feeds them to the application through the ISIS (scanner driver architecture) pipeline. This driver was configured to feed images at the maximum possible rate, which in most cases exceeded the speed of typical scanners.

The scenarios described below were tested under several variations of network conditions imposed on the connection between the web client and the web server, with the goal being to determine if and to what degree network latency could affect performance.

## Test Scenarios for Captiva Web Client

The following scenarios were used to test scanning rates, post processing time, and batch submission times under various configurations of CWC features, with and without use of ad hoc services (REST).

Unless otherwise noted, all batches consisted of 200 pages which were bitonal TIFF, 300 DPI, averaging 30 KB in size.

**Table 47. Captiva Web Client Scenario Descriptions**

| Scenario | Description |
| --- | --- |
| **Scenario 1**:<br><br>Plain | This was the simplest scenario of scanning 200 bitonal TIFF images and automatically inserting new document splits after every 4 pages, resulting in a batch structure of 50 documents with 4 pages each.<br>No ad hoc service (REST) calls were made. |
| **Scenario 2**:<br><br>Plain (large batch) | This was the same as Scenario 1 except batch size was 1000 pages, resulting in 250 documents. |
| **Scenario 3**:<br><br>Image Processing (REST) | This was similar to Scenario 1 except that each scanned image was sent to an ad hoc REST service to have Image Processing filters applied.<br>The filters used were<br>• Deskew<br>• Despeckle<br>• Overscan removal<br>• Auto crop |

| Scenario | Description |
|---|---|
| **Scenario 4**:<br><br>Barcode Detection | This was similar to Scenario 1 except that instead of using an arbitrary page counting rule to split documents, a barcode detection rule was used to split only when a barcode was detected on the image.<br>Since the barcode page repeated after every $4^{th}$ page, this resulted in the same document structure of 50 documents with 4 pages each.<br>Barcodes were detected within the browser session and did not require calls to ad hoc REST services. |
| **Scenario 5**:<br><br>HPA Classification + Zonal Extraction (REST) | In this scenario, the images were sent to ad hoc REST service to be classified and have data extracted. The first image in every group of 4 images was classified as an invoice and resulted in a new document split and extraction of invoice data. The remaining 3 images were treated as attachments without any data extraction.<br>The method of classification was "High Precision Anchor (HPA)" and the method of extraction was zonal OCR based on anchors. |
| **Scenario 6**:<br><br>Keyword Classification + FreeForm Extraction (REST) | This was the same as Scenario 5 except that the method of Classification was Keyword (which uses a full-page OCR technique) and the method of extraction was FreeForm (which also uses full-page OCR). |

## Benchmark Results for Captiva Capture Web Client

The tests performed for Capture Web Client fall into two usage categories: Scanning performance and batch completion/submission rates. All scenarios where tested with 10 Mbps bandwidth to the web server, and round trip latencies of 0 ms, 50 ms, and 150 ms.

Note: "Round trip latency" is the net combined latency for sending and receiving. It is the latency value reported when performing a "ping" test to the web server.

## Scanning Performance Results

On the following page are the results of measuring the "real time" scanning rate under different conditions. For these tests, a simulated scanner driver was used. This driver emits pages at the fastest possible rate, being throttled only by the application's ability to consume them. Thus, the rates shown below are the theoretical maximums for the given hardware.

Since the maximum scanning rate gradually decreases as the batch becomes larger, the measured real time scanning rates are repeated in the table at each 200 page interval.

When a real scanner is used, the scanning rate will be limited by either the scanner itself or the application, depending on which has the lower scanning rate.

- If the theoretical maximum was 350 IPM (images per minute) and the scanner's rate was 100 IPM, the actual scan rate would be 100 IPM
- If the theoretical maximum was 80 IPM, then the actual scan rate would be 80 IPM, despite the scanner being capable of 100 IPM

**Table 48. Scan Performance Results**

| **Scenario** | **Image Per Minute (IPM) Rate** *at different round trip latencies* | | |
|---|---|---|---|
| | *@ 0 ms* | *@ 50 ms* | *@ 150 ms* |
| **Scenario 1** (Plain) | | | |
| **START** | 450 | 430 | 420 |
| **At 200 pages** | 330 | 330 | 330 |
| **Scenario 2** Plain (large batch) | | | |
| **START** | 360 | 350 | 340 |
| **At 200 pages** | 280 | 280 | 280 |
| **At 400 pages** | 170 | 170 | 170 |
| **At 600 pages** | 110 | 110 | 110 |
| **At 800 pages** | 80 | 80 | 80 |
| **At 1000 pages** | 60 | 60 | 60 |
| **Scenario 3** Image Processing (REST) | | | |
| **START** | 400 | 420 | 420 |
| **At 200 pages** | 300 | 310 | 320 |
| **Scenario 4** Barcode Detection | | | |
| **START** | 150 | 150 | 150 |
| **At 200 pages** | 140 | 140 | 140 |
| **Scenario 5** HPA Classification + Zonal Extraction (REST) | | | |
| **START** | 430 | 440 | 440 |
| **At 200 pages** | 230 | 240 | 270 |
| **Scenario 6** Keyword Classification + FreeForm Extraction (REST) | | | |
| **START** | 450 | 440 | 450 |
| **At 200 pages** | 350 | 360 | 360 |

## Conclusions on Scanning Performance

**Effect of Network Latency:**

Network latency has a negligible effect on actual scanning rate.

**Effect of Batch Size:**

Since the scanning rate decreases as batch size increases, there is potential for some degradation in scanning performance with very large batches. The point at which the degradation would be seen depends on the speed of the scanner, with faster scanners being more likely to see some degradation after about 600 pages. With low speed scanners, the operator will most likely not notice any difference in scanning speed throughout the entire scanning operation.

**Effect of Barcode Detection:**

The use of barcode detection will reduce the maximum theoretical scanning rate. For most typical scanners this will not be noticeable. It may be noticeable with production level scanners or when importing images.

**Effect of ad hoc services (REST services):**

The use of ad hoc services for Image Processing or Classification/Extraction had only a minor impact on scanning rate which will rarely be noticed.

Example 1: In scenario 4 (barcode detection), the maximum scanning rate started at 150 IPM and dropped to 140 IPM after 200 pages. If a scanner rated for 100 IPM were used, the actual scan rate would remain at a constant 100 IPM through the batch. However, in the unlikely event that an extremely high speed scanner such as 180 IPM was used, the actual scan rate would be constrained to approximately 140 to 150 IPM.

Example 2: In scenario 2 (plain, large batch) below, the maximum scanning rate started at 360 IPM and dropped to 110 IPM after 600 pages and to 80 IPM after 800 pages in a single batch. If a scanner rated for 100 IPM were used, the actual scan rate would remain at a constant 100 IPM for batches up to about 650-700 pages. For batches larger than that, the actual scan rate would begin to decline slightly as the batch size grew beyond 700 pages.

## Batch Completion and Submission Performance

The total time required to scan and submit a batch can be broken into 3 phases:

1) Scanning

2) Waiting for any post processing to complete

3) Submitting

The first set of results for scanning (on the previous page) showed that network latency was not a significant factor for scanning rates.

When ad hoc services are used (REST service calls), however, there are round trip operations made to/from the web server for each page scanned. Each round trip incurs a penalty from network latency.

Additionally, when a batch is submitted, it is broken into “chunks” and uploaded in pieces. This “chunking” also incurs some penalty from network latency.

The results on the following page are for the same scenarios previously discussed and illustrate the amount of time spent in each of the 3 phases of batch scanning under different network latencies.

**Table 49. Scan / Post Processing / Submission Performance Results**

| **Scenario** | **Scan, Post Scan, and Submit Durations (in seconds)** *at different round trip latencies* | | |
|---|---|---|---|
| | *@ 0 ms* | *@ 50 ms* | *@ 150 ms* |
| **Scenario 1** (Plain) | | | |
| **Scan Duration** | 31.0 | 31.1 | 31.2 |
| **Post Processing** | 5.5 | 5.5 | 5.6 |
| **Submit Duration** | 14.0 | 17.9 | 26.2 |
| **Scenario 2** Plain (large batch) | | | |
| **Scan Duration** | 477.9 | 485.8 | 494.7 |
| **Post Processing** | 8.6 | 9.6 | 10.9 |
| **Submit Duration** | 219.8 | 278.9 | 337.1 |
| **Scenario 3** Image Processing (REST) | | | |
| **Scan Duration** | 34.2 | 33.1 | 32.5 |
| **Post Processing** | 61.1 | 70.0 | 88.9 |
| **Submit Duration** | 13.5 | 18.3 | 27.1 |
| **Scenario 4** Barcode Detection | | | |
| **Scan Duration** | 83.9 | 83.5 | 83.5 |
| **Post Processing** | 5.2 | 5.2 | 5.2 |
| **Submit Duration** | 13.1 | 17.9 | 26.3 |
| **Scenario 5** HPA Classification + Zonal Extraction (REST) | | | |
| **Scan Duration** | 40.2 | 39.2 | 35.1 |
| **Post Processing** | 20.4 | 26.3 | 40.5 |
| **Submit Duration** | 18.3 | 22.8 | 32.5 |
| **Scenario 6** Keyword Classification + FreeForm Extraction (REST) | | | |
| **Scan Duration** | 30.4 | 31.1 | 30.3 |
| **Post Processing** | 263.7 | 273.3 | 294.6 |
| **Submit Duration** | 26.5 | 31.9 | 42.7 |

## Conclusions on Batch Completion and Submission Performance

**Effect of Network Latency:**

Network latency has a negligible effect on actual scanning rate, and a modest effect on post processing wait time, but only in cases where ad hoc services are used. When no ad hoc services are used, there is negligible effect on post processing wait times due to latency.

The area of batch submission is where the most noticeable impact of network latency can be seen. With 150 ms latency (this would be on the order of an overseas connection) the batch submission time could be twice as much as when there is no latency.

For smaller network latencies like 50 ms (perhaps from central US to the east coast) there is roughly a 25% increase in the time required to submit a batch as compared with no latency.

**Effect of Batch Size:**

Similar to scanning rates, the overall batch submission rate declines as batch size increases. If possible, avoid creating very large batches.

**Effect of Ad Hoc Services (REST):**

When ad hoc services are used during the scanning process, they begin to run in parallel with scanning, as soon as the first images are acquired. While they do not impact the rate of scanning significantly, once scanning is complete any remaining ad hoc service calls must be completed before the batch is ready to be reviewed or submitted by the user. This time after scanning is complete is referred to as the "post scan" time in this document. The amount of time here will vary depending on batch size and the nature of the REST call, as well as on network latency.

Scenarios 3, 5 and 6 used ad hoc services in these tests. In Scenarios 3 and 6, the post processing time appears to be significantly greater than the actual scan time, but it must be understood that the scan time here is artificially fast due to the use of a simulated scan driver running at an unthrottled speed.

If we look at the worst case example of Scenario 6, the sum of scan time and post processing time was about 294 seconds. This represents the total time spent on ad hoc services. If the scanner had been rated at 40 images per minute, the entire scan time would have been about 300 seconds, and therefore all of the ad hoc service calls would have presumably completed at the same time as the scanning. With faster ad hoc services, such as using HPA classification and zonal extraction, even higher scanner speeds could be used with little or no post scanning delay.

The take away from this is that slow ad hoc services have the potential to cause some delay after scanning, but mainly in cases where the ad hoc services are unusually slow or the scanner is very fast. In these cases, additional improvements could be made on the REST servers (such as adding more module servers) to increase their throughput and potentially reduce this post scanning delay. See recommendations for the REST server deployment.

## General Sizing Guidelines

Number of REST Servers

- Start with 2 REST Servers for the first 10 concurrent users

- Add an additional REST server for each 40 additional concurrent users

Number of Module Servers

- Determine the number of client module instances (Image Processor, Image Converter, Classification, Extraction, NuanceOCR) required using the client module sizing guidelines in this document
- The module which requires the largest number of instances determines the baseline number of Module Servers required
- For example, if it is determined that
    - Classification requires 8 instances
    - Extraction requires 7
    - Image Processor requires 4
    - Image Converter requires 6
- Then 8 module server s would be required as a baseline
- To improve real time responsiveness under concurrent loads, add about 25% more
- Start with 10 module servers (8 + 25%), and adjust if necessary after concurrency testing
- Configure each module server to run 4 instances of each module (this is done with the “REST Service Config” (Captiva REST Server Configuration) utility

Hardware recommendation

- REST Server: 4 to 8 CPUs, 8 GB or more RAM
- Module Server: 4 CPUs, 32 GB RAM

# Chapter 6 Captiva Administrator

## Introduction

Captiva Administrator is an administration tool for Captiva Capture that enables administrative personnel to monitor, edit, adjust, and respond to most administrative needs. Captiva Administrator centralizes and consolidates many activities, including the following key administrator activities:

- Server administration
  - Licensing, configuring, monitoring, and controlling all InputAccel Servers within your organization
  - Licensing, establishing, and monitoring ScaleServer groups of InputAccel Servers
- Client module license administration
- User administration
- Client administration
- Reporting, logging, and performance monitoring administration
- Configuring access control through a system of users, groups, permissions, and access control lists
- Web Services configuration and administration

This section describes the testing methodology used to test Captiva Administrator and provides the benchmark results and sizing recommendations to improve the performance of Captiva Administrator.

The benchmark testing results may vary for your specific production environment. Use the results and recommendations presented in this document as a starting point to help you determine the appropriate performance and sizing for your specific production environment.

# Test Environment and Methodology

## Method of Testing

Performance tests for Captiva Administrator consisted of testing three types of screen sets; screens that contained large data sets, screens that contained small data sets and settings screens. The large data sets screen testing included eight screens such as Batch Traffic and Logs. The small data set screens set contained twelve screens such as Report Definitions, Module Licenses, and Processes. The Settings screen set contained twelve screens such as Configured Purge Details, Role Definition, and New Batch.

The performance testing for Captiva Administrator measured the load time of Captiva Administrator pages given a certain amount of data in the system. Captiva Administrator and client modules were installed on the same machine as described in . The InputAccel Server and InputAccel Database were installed together on a separate system as described in *Physical Machine Configuration Used for Database Testing*.

## Test Data

The system had the following data for testing performance:

- Number of Servers: 10
- Batches: 25000
- Processes: 120

- License Policy: 140
- Modules: 95
- Logs: 46330

## Captiva Administrator Benchmark Results

The following are the benchmark results:

- For screens that display large data sets, 60% of screens loaded within a time of 3 seconds or less. Screens with large data sets included:
  - Batch Properties
  - Batch Traffic
  - Logs
  - Process - IA Values
  - Process - Indexed IA Values
- For screens that display small data sets, 75% of screens loaded within 2 seconds or less. Screens with small data sets included:
  - Column Manager
  - Image Viewer
  - License Codes
  - Log Details
  - Log View Filters
  - Report Definitions
  - Module Licenses
  - Server Activations
  - Departments
  - Module Connections
  - Processes
  - Servers

The test results can be extrapolated for other small data set screens such as Purge Definitions, Modules, Purges, and Reports.

- For screens that did not display grids, 100% of screens loaded within 1 second or less. Settings screens included:
  - Move Batch
  - Global Options
  - Purge Definition Settings

- Log Rule Filter Definition
- Reports Welcome
- Role Definition
- Sink Definition
- Grid Selector
- Copy Setup Value
- Process — Add Batch
- Upgrade Process
- Process Settings

The test results can be extrapolated for other screens that do not contain grids such as Report Definition Settings, Context Menu, and Print Pop Up.

## Captiva Administrator Critical Factors and Sizing Recommendations

The following critical factors and sizing considerations can affect Captiva Administrator performance:

- The response time for loading a page is impacted by the amount of data that is retrieved from the database. The more data retrieved, the slower the response time. This factor is also page dependent, because each page displays varying amounts of data. The response time for loading a page can be improved by reducing the number of rows displayed in a grid. Set this option in the Options › Default Settings pane in Administration Console.
- The recommended hardware requirement is listed in . It is recommended that the browser is run on a fast system (3 GHz Xeon CPU, with 2-3GB of RAM, for example) to improve page loading time.
  - Client modules can be run on the same machine as the AC web browser system, as long as they are not memory intensive modules. Memory intensive modules will affect the load time of AC page

# Chapter 7 Components over a WAN Network

Performance of Captiva Capture components separated from other components by a WAN emulator was benchmarked. The following section provides benchmark results and recommendations.

## InputAccel Database over WAN

In this scenario, the machine running the SQL Server that hosts the InputAccel Database is separated from all other components by a WAN emulator.

This deployment is not recommended for production although it may yield acceptable performance if the following conditions are met:

1. All Reporting and Audit log rules are disabled. When these rules are enabled, data written to the InputAccel Database significantly reduces InputAccel Server and client module throughput.
2. A minimum of 50 Mbps (mega-bits per second) bandwidth and maximum of 25 millisecond round-trip latency is required for the WAN.
3. Low requirements for page throughput.

When these conditions are met, expect a drop in throughput of about 50% in comparison to a deployment that uses a LAN. EMC recommends that you thoroughly test this deployment for your production environment and benchmark the performance accordingly.

## Captiva Completion over WAN

For these scenarios, the machine running Captiva Completion was separated from the InputAccel Server by a WAN emulator. Each of the scenarios has been tested with the following WAN settings:

- Transfer rate: 50 Mbps
- Round trip latency to InputAccel Server (as per “ping”)

— 0 ms

— 50 ms

— 150 ms

### Benchmark Results

Scenarios for Captiva Completion addressed 3 different data sizes and 2 trigger levels.

Data Sizes

- 100 fields (119 KB of UIM data)
- 320 fields (458 KB of UIM data)

    - 20 primary fields +
    - Table with 50 rows and 6 columns
- 1000 fields (2096 KB of UIM data)
    - 40 primary fields +
    - Table with 80 rows and 12 columns

Trigger Levels

- Document (each task is a single document with 5 pages)
- Batch (each task is a group of 50 documents with 5 pages per document)
  NOTE: For batch level tests, the total UIM data transferred was 50x greater due to 50 documents in the task

The measured result was the "task to task" time (in seconds) that a user would experience. This delay is the net sum of sending current task's data back to the InputAccel Server and receiving the next task's data. This measurement represents the total delay experienced by the user while waiting for the UI of the next task to become available after finishing a similar task.

**Table 50. Captiva Completion over WAN Scenario Results**

| Scenario Name | Trigger Level | Field Data | Total UIM task data | Task to task time (sec) at various latencies | | |
|---|---|---|---|---|---|---|
| | | | | 0 ms | 50 ms | 150 ms |
| Scenario 1: Doc Level Scalar Values | 1 Document | 100 fields | 0.12 MB (119 KB) | 1.0 | 3.2 | 9.6 |
| Scenario 2: Doc Level Medium Table | 1 Document | 320 fields | 0.47 MB (457 KB) | 1.7 | 4.1 | 11.4 |
| Scenario 3: Doc Level Large Table | 1 Document | 1000 fields | 2.0 MB (2,096 KB) | 4.1 | 7.0 | 18.1 |
| Scenario 4: Batch Level Scalar Values | 7 Batch (50 docs) | 500 fields (50 x 100) | 5.8 MB (50 x 119 KB) | 3.8 | 8.3 | 21.9 |
| Scenario 5: Batch Level Medium Table | 7 Batch (50 docs) | 16,000 fields (50 x 320) | 22 MB (50 x 457 KB) | 6.6 | 29.9 | 90.4 |
| Scenario 6: Batch Level Large Table | 7 Batch (50 docs) | 50,000 fields (50 x 1000) | 102 MB (50 x 2,096 KB) | 17.5 | 149.7 | 447.4 |

## Summary of Results

Extremely large quantities of task data such as 22 MB or 102 MB in Scenarios 5-6 can result in extremely long wait times for the user under WAN conditions. Possible remedies for this situation include

- Trigger at document level whenever possible for WAN conditions
- If batch level is required, minimize the number of documents per batch, or reduce the amount of data (if possible) per document

Additional factors were tested but not reported in the table above. These factors proved to *not have a significant impact* on task to task performance. These included

- Outputting dynamic values to another step
- Writing report data to the database

Thumbnail display also did not t impact significantly the user waiting time between tasks when latency is high because thumbnail retrieval and/or generation occurs in the background. However, the time required for all thumbnail images to be rendered can be significant when thumbnail files do not already exist for the images. In this case, Captiva Completion must retrieve the all of the images in their entirety in order to generate thumbnails. When pre-generated thumbnails exist, Captiva Completion need only retrieve a set of very small thumbnail files. Thumbnails are generated by the following modules

- ScanPlus
- Image Processor
- Image Converter

It is recommended to route the "OutputImage" from one of the above modules to Captiva Completion to ensure that the thumbnail files will exist in advance.

## Critical Factor

- Trigger Captiva Completion at level 1 rather than level 2 or 7 when folders contain more than 10 documents. When triggered at level 2 or higher, a significantly greater amount of document data is retrieved or written all at once which significantly increases the user waiting time between tasks.
- The amount of UimData in a document type impacts the user waiting time significantly. The fewer UimData fields the document type contains, the less is the user waiting time.

## Non-Critical Factors

- Document resources are downloaded to the client machine only once, and are updated only if the file version on the server is different than the version on the client machine.
- Outputting dynamic values to another step and reporting data to the database does not cause any significant delay to the user task-to-task time.
- Thumbnail generation does not impact significantly on the user waiting time between tasks when latency is high. The thumbnail size does not impact the user waiting time.

# ScanPlus over WAN

The ScanPlus-over-WAN benchmark environment is set up as follows:

- The machine running ScanPlus is separated from the InputAccel Server and the InputAccel Database by a WAN emulator.
- Network bandwidth is fixed at 50 Mbps (which is approximately OC-1 speed).
- All benchmark scenarios used 23K images and 2.3 MB batches.

**Note:**

- The ScanPlus module connects with the InputAccel Server using TCP/IP, not HTTP.
- The round-trip latencies in these performance benchmarks are the same latencies that would occur when a ping is sent from the ScanPlus client machine to the InputAccel Server.

The following ScanPlus performance benchmarks were performed:

- Scenario 1: ScanPlus Module Startup Time
- Scenario 2: Duration of Process Selection and New Batch Creation
- Scenario 3: Average Time to Create, Scan, and Close a Batch

## Scenario 1: ScanPlus Module Startup Time

ScanPlus module startup time is defined as starting when the user starts the module from the Start menu shortcut (configured with automatic login credentials) and ending when the Create Batch button is enabled.

During the ScanPlus module startup stage, WAN latency affects network communication of the following processes:

- Initial module connection to InputAccel Server
- Authentication of the user
- Retrieval of permissions and log rules
- Client scripting DLL

**Note:** Server processes and batches are not part of the module startup process.

**Table 51. Scenario 1: Module Startup Time**

| | Latency (Milliseconds) | | | |
|---|---|---|---|---|
| | 0 | 50 | 100 | 150 |
| Total Module Startup Time (Seconds) | 7.2 | 9.7 | 12.8 | 15.0 |

## Scenario 2: Duration of Process Selection and New Batch Creation

When the Create Batch button is pressed, ScanPlus retrieves a list of process names from the InputAccel Server and, optionally, a list of batch names that are already on the server. Depending on the number of processes and batches, this retrieval can result in a significant amount of data being retrieved from the server, which, in turn, can result in a delay before a new batch can be created.

During this retrieval stage, WAN latency affects network communication for the retrieval of process and batch lists.

To highlight the impact of the number of processes and batches on the duration of process selection and new batch creation, the benchmark environment is set up to test a variable number of processes and batches (in addition to the setup specified in *ScanPlus over WAN*).

**Table 52. Scenario 2: Duration of Process Selection and New Batch Creation**

| | | Round-trip Latency (Milliseconds) | | | |
|---|---|---|---|---|---|
| | | 0 ms | 50 ms | 100 ms | 150 ms |
| | 1 Process<br>0 Batches | 0.60 | 3.0 | 5.4 | 7.6 |

| | | Round-trip Latency (Milliseconds) | | | |
|---|---|---|---|---|---|
| Duration of Process Selection and New Batch Creation (Seconds) | 100 Processes<br>0 Batches | 0.9 | 22.8 | 43.7 | 65.3 |
| | 100 Processes<br>15000 Batches<br>(Fetch batch names enabled) | 2.5 | 25.0 | 48.6 | 70.2 |
| | 100 Processes<br>15000 Batches<br>(Fetch batch names disabled) | .8 | 22.8 | 44.9 | 67.7 |

## Scenario 3: Average Time to Create, Scan, and Close a Batch

Scanning can start after a process is selected, a batch name entered, and a new batch created. While scanning, pages are transmitted to the InputAccel Server over the WAN and a confirmation for each page is sent back to the ScanPlus module. After the batch is closed, the cycle repeats, and ScanPlus retrieves the process list again so that the user can select the same process or another process for the next batch.

At this batch creation and scanning stage, WAN latency affects the network communication of the following processes:

- Transmission of images to the InputAccel Server
- Creation of new document nodes
- Interaction between ScanPlus and server with regard to creating or closing the batch
- The subsequent retrieval of the process list after each batch completes

The average time for single batch creation, scanning, and closure is measured by repeatedly creating 40 batches and then dividing the total time by 40.

In addition to the setup specified in *ScanPlus over WAN*, this benchmark environment is set up as follows:

- 40 batches are repeatedly created with the Auto Batch Creation feature of ScanPlus.
- Each batch contains 100 bitonal images organized into 10 documents of 10 pages each.
- After each batch is closed, the process list (containing 10 processes) is retrieved again before the next batch is created.
- The scanning is performed with a simulated scanner configured to scan at 600 images per minute so that any performance degradation from WAN latency can be fully attributed to the WAN rather than to the scanner.

**Note:** With a slower scanner, WAN degradation might not be that noticeable because the slower scanning speed would compensate for the longer data transmission times over the WAN. With a greater number of processes, the delay between each batch might increase (see the results from Scenario 2), but the actual time spent scanning would not be affected.

**Table 53. Scenario 3: Average Time to Create, Scan, and Close a Batch of 100 Pages, Split into 10 Documents**

| | Round-trip Latency (Milliseconds) | | | |
|---|---|---|---|---|
| | 0 ms | 50 ms | 100 ms | 150 ms |
| Average Time (Seconds) | 3.7 | 21.3 | 39.8 | 58.0 |

## Recommendations

Running ScanPlus over a WAN is not recommended but might produce acceptable performance under the following conditions:

- The sizes of the scanned images are small.
- The scanner is not extremely fast.
- The WAN round-trip latency is less than 50ms.
- The number of processes on the InputAccel Server is relatively low.

In addition, EMC recommends the following:

- Disable the ScanPlus setting Fetch existing batches list on the New Batch screen to potentially improve performance.
- Use ScanPlus version 7.5 or greater because version 7.5 includes optimizations that make WAN scanning speeds up to 25% faster than earlier ScanPlus versions.

These are general guidelines. Refer to the results from benchmark testing to determine if scanning over a WAN results in acceptable performance in your environment.

# Captiva Identification over a WAN

For these scenarios, the machine running Captiva Identification was separated from the InputAccel Server by a WAN emulator. Each of the scenarios has been tested with the following WAN settings:

- Transfer rate: 50 Mbps
- Round trip latency to InputAccel Server (as per "ping")
  - 0 ms
  - 50 ms
  - 150 ms

## Benchmark Results

Scenarios for Captiva Identification addressed 2 recognition projects (DPPs) containing different numbers of templates. Trigger level 7 was used in all cases. No pre-indexing field was used.

The measured result was the “task to task” time (in seconds) that a user would experience. This delay is the net sum of sending current task’s data back to the InputAccel Server and receiving the next task’s data.

The first batch processed will incur an additional delay to load the recognition project into memory, but once loaded the subsequent batches do not need to load the project again. Because this one-time delay occurs only on the first batch, that batch was excluded from the results below. The task-to-task times shown below are valid for the 2nd and all subsequent batches processed.

The test batches used consisted of 50 documents with 5 pages each, for a total of 250 pages in each batch. The pages did not have any existing classification data yet.

**Table 54. Identification over WAN Scenario Results**

| Scenario Name | Trigger Level | Task to task time (sec) at various latencies | | |
|---|---|---|---|---|
| | | 0 ms | 50 ms | 150 ms |
| Scenario 1: 500 Templates | 7 Batch | 3 | 61 | 179 |
| Scenario 2: 3,000 Templates | 7 Batch | 3 | 61 | 179 |

## Summary of Results

- Once the recognition project (DPP) is loaded into memory after the first batch, its size (e.g. number of templates) plays no role in task to task performances. The results for 500 templates and 3000 templates are identical.

- Network latency has a significant impact on Identification performance. If Identification must be run over WAN connection, reduce the number of pages in each batch to help with batch to batch performance.

## Critical Factors

- Keep batch sizes small when using Captiva Identification over a WAN.

## Non-Critical Factors

- DPP size / number of templates was not a critical factor once initially loaded in the first task.

- Outputting dynamic values to another step and reporting data to the database does not cause any significant delay to the user task-to-task time.

- Thumbnail generation does not impact significantly on the user waiting time between tasks when latency is high. The thumbnail size does not impact the user waiting time.

# Chapter 8 Appendix

## Physical Machine Configuration Used for Database Testing

**Table 55. System Properties for Server and Database Machines**

| Manufacturer | Dell Inc. |
|---|---|
| Model | PowerEdge 2950 |
| CPU Cores | 8 CPUs x 2.826 GHz |
| Processor Type | Intel Xeon CPU E5440 @ 2.83 GHz (Quad core) |
| Processor Sockets | 2 |
| Cores per Socket | 4 |
| Logical Processors | 8 |
| Hyperthreading | Inactive (not supported on this CPU type) |
| RAM | 32 GB |
| Disk Controller | PERC6/i |
| Disk Drives | 15K SAS |
| Operating System | Windows Server 2008 R2 (64-bit) natively installed |

## Machine Configuration for Server: 64-bit Improvements and .NET-based XPP Definitions

**Table 56. Base Hardware Configuration for Server 64-bit Testing**

| Manufacturer | Cisco |
|---|---|
| Model | Cisco UCSC-C240-M4SX |
| CPU Cores | 24 CPUs x 2.30 GHz |
| Processor Type | Intel® Xeon® CPU E5-2670 v3 @ 2.30 GHz |
| Processor Sockets | 2 |
| Cores per Socket | 12 |
| Logical Processors | 48 |
| Hyperthreading | Active |
| Memory | 256 GB RAM [2133 MHz] (128 GB per CPU |
| Disk Controller | Cisco 12G SAS with 1 GB cache (MegaRAID SAS Invader) |

| Manufacturer | Cisco |
|---|---|
| Disk Array for VMs | 12x 600 GB SAS drives (10K RPM) in single RAID 6 array |

**Table 57. Virtual Machine Configuration for Server 64-bit Testing**

| Virtualization Vendor | VMware |
|---|---|
| Version | ESXi 5.5.0 (2403361) |
| Virtual CPUs | 24 virtual sockets (1 core per socket) |
| Allocated RAM | 230 GB |
| Boot VMDK | HD1 (Drive C) on virtual SCSI-0 (LSI Logic SAS) |
| Data VMDK | HD2 (Drive D) on virtual SCSI-1 (Paravirtual) |
| Operating System | Windows Server 2012 R2 |

**Note:** both VMDK files were on same local datastore of 12-drive RAID 6 array. All resources configured with no reservations or limits, and "Normal" shares

# Machine Configuration Used for Server: Virtualized Benchmark Testing

**Table 58. Physical Machine Configuration Used for Server: Virtualized Benchmark**

| Manufacturer | Dell Inc. |
|---|---|
| Model | PowerEdge 2950 |
| CPU Cores | 8 CPUs x 2.826 GHz |
| Processor Type | Intel® Xeon® CPU E5440 @ 2.83 GHz (Quad core) |
| Processor Sockets | 2 |
| Cores per Socket | 4 |
| Logical Processors | 8 |
| Hyperthreading | Inactive (not supported on this CPU type) |
| RAM | 32 GB |
| Disk Controller | PERC6/i |
| Disk Drives | 15K SAS |
| Operating System | Windows Server 2008 R2 (64-bit) natively installed |
| Modifications | Windows was limited to 4 CPUs and 8 GB RAM by using following commands: bcdedit /set {current} removememory 24576 bcdedit /set {current} numproc 4 |

**Note:** Windows booted from a 100 GB C: partition on a Disk 1 under a PERC6/I controller, RAID 10 array. InputAccel Server data was stored on a 100 GB D: partition on same Disk 1.

**Table 59. Virtual Machine Configuration Used for Server: Virtualized Benchmark**

| Manufacturer | Dell Inc. |
|---|---|
| Model | PowerEdge 2950 |
| CPU Cores | 8 CPUs x 2.826 GHz |
| Processor Type | Intel® Xeon® CPU E5440 @ 2.83 GHz |
| Processor Sockets | 2 |
| Cores per Socket | 4 |
| Logical Processors | 8 |
| Hyperthreading | Inactive (not supported on this CPU type) |
| RAM | 32 GB |
| Disk Controller | PERC6/i |
| Disk Drives | 15K SAS |
| Operating System | ESXi 4.1.0.260247 |
| Virtual Machine | Windows Server 2008 R2 (guest OS)<br>4 Virtual CPUs<br>8 GB RAM<br>VMXNET 3 virtual network adapter<br>LSI Logic SAS and VMware Paravirtual SCSI adapters |

## Recommendations for the Environment Used for SQL Server

The following recommendations are for SQL Server 2008, 2008 R2, or 2012 hosting only the InputAccel Database:

**Table 60. SQL Server Environment**

| Captiva Capture (tasks per Hour) | Recommended SQL Server Edition | CPU Core | Disk System |
|---|---|---|---|
| Low (‹ 50,000) | Without Reports: Express<br>With Reports: Standard | 2 | Standard Disks |
| Medium (50,000 - 400,000) | Without Reports: Standard (x64)<br>With Reports: Enterprise (x64) | 2–4 | RAID 5 or 10 with R/W Caching |
| High (› 400,000) | Enterprise (x64) | 4+ | Hardware RAID 10 with R/W Caching |

**Note:**

- For Low Task Volume, SQL Server Express edition is limited to a maximum database size of 4 GB for SQL Server 2008 and 10 GB for SQL Server 2008 R2 and SQL Server 2012.. Therefore, it is viable for hosting the InputAccel Database only if you are sure your database will never grow larger than that. If you use reports or audits, you will almost certainly exceed this limit; therefore, SQL Express should be used only in low volume environments that do not use reporting or audit log rules.
- For Medium Task Volume, x64 editions of SQL Server are generally recommended because they have significant advantages in memory availability.
- For Low Task Volume without Reports, the file-based database (no SQL Server) is also a viable option.

## Test Environment Used for Testing of ODBC Export

**Table 61. System Properties for ODBC Export Machines**

| Item | Required |
|---|---|
| Hardware | Dell Optiplex 755<br>• Intel Core2 Duo E6850 @ 3.00 GHz (dual core)<br>• 3.25 GB RAM<br>• 7200 RPM SATA II drive<br>• 1 Gbit Network interface card |
| Operating System | Windows 7 (32-bit) |
| Software | McAfee VirusScan Enterprise 8.5i |

## Test Environment Used for Testing of Client Modules and Administrator

**Table 62. System Properties for Capture Client Modules and Administrator**

| Item | Required |
|---|---|
| Hardware | Dell Optiplex 780<br>• Intel Q9650 @ 3.00 GHz (quad core)<br>• 4 GB RAM<br>• 7200 RPM SATA II drive<br>• 1 Gbit Network interface card |
| Operating System | Windows Server 2008 R2 Enterprise (64-bit) |

# Test Environment Used for Testing of Classification and Extraction Modules

**Table 63. System Properties for Capture Client Modules and Administrator**

| Item | Required |
|---|---|
| Hardware | Lenovo C30 Workstation<br>• Intel Xeon E5-2609 v2 @ 2.5 GHz (quad core)<br>• 4 virtual CPUs allocated to VM<br>• 4 GB RAM allocated to VM<br>• 1 Gbit Network interface card |
| Operating System | vSphere ESXi 5.5 as virtual machine host<br>VM image built with Windows Server 2012 R2 |

# Explanation of Columns in Client Module Benchmark Results

The *Client Module Recommendations* section discusses benchmark results of client modules and recommendations to size the modules.

The benchmark results table lists results in various columns. This section explains the columns used in the benchmark results tables.

- **Client Module % Processor Time**: This value comes from the Windows **Processor% Processor Time** performance counter for the module's process. It is the percentage of elapsed time that all the process threads used to execute instructions.
- **Average CPU % Utilization for Both Cores**: This column shows the average total CPU utilization for all cores. This value comes from Windows**Processor(_Total)\% Processor Time** performance counter . The maximum this value can be is 100%, even on multi-core machines.
- **Processor <n> % utilization**: This is the % processor time for the indicated CPU core. This value comes from Window's **Processor(0)\% Processor Time and Processor(1)\% Processor Time**. The maximum this value can be is 100%.
- **Disk Usage Read / Write per Second**: This is the average number of bytes transferred from a disk during read/write operations. These values comes from Windows **Disk Read Bytes/sec** and **Disk Write Bytes/sec** performance counters.
- **Network Data Received / Sent per Second**: This is the number of bytes of data sent to and received from the network interface. These values come from Windows **Network Interface : Bytes Received/sec.** and **Network Interface : Bytes Sent/sec.** performance counters.
- **Private Bytes**: The number of bytes exclusively allocated to the module process and that cannot be shared with other processes. This value comes from Windows **Process Private Bytes** performance counter.
- **CPU % Processor Time**: The average percentage of elapsed time that the processor spends to execute a non-idle thread. This values comes from Windows **Processor\% Processor Time** performance counter.

- **Effective Units of Full Module Performance:** Processing capability of running additional instances of the module on a multi-core machine. For example, running 4 instances of NuanceOCR on a single Quad-Core desktop machine yielded processing capability equivalent to 3.78 copies of NuanceOCR on four separate machines.
- **Number of IA Value Requests (per second) from Server:** The number of IA Values requested by the InputAccel Server for export.

# Image Sets and Settings Used

## NuanceOCR

The NuanceOCR testing scenario used a multipage image file containing 109 pages. Nine of those pages were patch codes and were discarded after being imported into ScanPlus. The 100 images that NuanceOCR processed were 300 dpi binary images.

### Settings Used for the Benchmark Testing

#### Document Recognition Settings

Auto-rotate image before recognition: No

Recognition Language: English

Spelling Language: English

Enable automatic spelling correction: No

Code page: Automatic (1252)

#### Zone Definition Settings

Enable OCR-assisted indexing: No

Selected character filters: None

Recognition engine: Automatic

Trade off: ACCURATE (unless otherwise specified)

Filling method: Machine Print (OMNIFONT)

#### Optical Mark Recognition

Frames: Automatic

Sensitivity: Normal

## Classification

Images for scenario 1 - 4: A batch of 100 binary images (300 dpi), around 36 KB per image (total size 3.6 MB). The 100 images were classified at more than 90%, and separators were used to create 10 documents of 10 pages each.

Images for scenario 5: A batch of 100 images (300 dpi) was used (total size 5.5 MB).

Images for scenario 6: A batch of 100 colored images (200 dpi) was used, around 690 KB per image (total size 69 MB).

## Automatic - HPA Template and Keyword (Passive Template) Scenario Image Set

The image sets in this section were used to test the following scenarios:

- Automatic + HPA 5 template image set
- Automatic + HPA 500 template image set
- Automatic + HPA 3000 template image set
- Keyword (Passive Template) classification image set

ENROLLMENT FORM &
YEARLY COORDINATION OF BENEFITS AND DEPENDENT STATUS STATEMENT

ATTENDING PHYSICIAN'S SUPPLEMENTARY STATEMENT


Input Accel
for Invoices

INVOICE
Separator

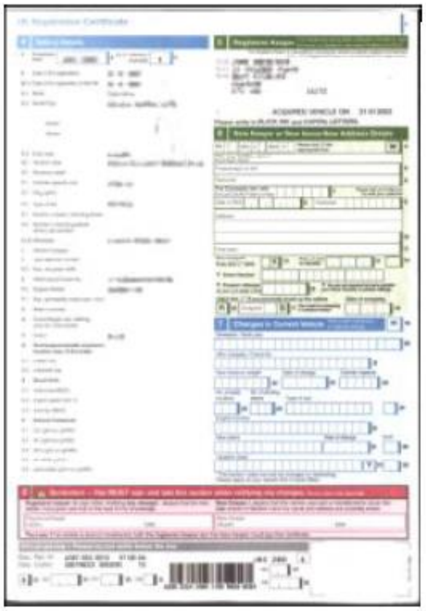
CAPTIVA


FONGECIF


Yokis

Facture N°

# Automatic + HPA Colored Image Sets



TITRE DE PROPRIÉTÉ

2000

BON DE COMMANDE

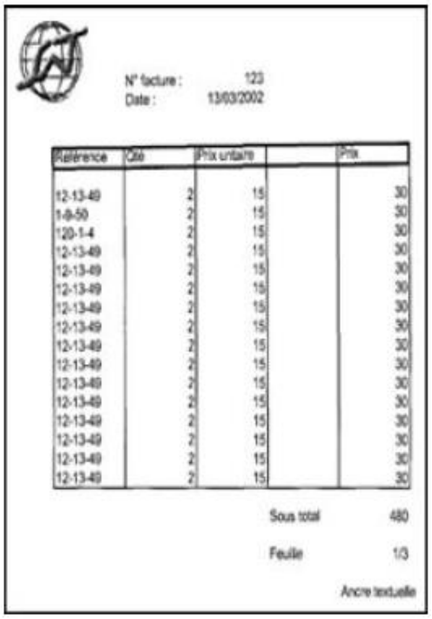
GRATUIT GRATUIT

Young's Market Company

Chili's #171
Pleasanton

TOTAL

## Text Matching Classification Scenario Image Set



N° facture : 123
Date : 13/03/2002

| Référence | Qté | Prix unitaire | | Prix |
|---|---|---|---|---|
| 12-13-49 | 2 | 15 | | 30 |
| 1-8-50 | 2 | 15 | | 30 |
| 120-1-4 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |
| 12-13-49 | 2 | 15 | | 30 |

Sous total 480

Feuille 1/3

Ancre textuelle

## Extraction

FACTURE

COPI CABANA/STUDIO G
2109 E. 27TH STREET
VERNON, CA 90058-1103




## PAL Supervisor

A set of 1,000 images was used and copied multiple times to get a collector of 5000, 10000, 20000, 40000, 60000, and 120000 images.

In each project, the minimum number of documents required to create a template was set to 5. So even though some images are identical we expect roughly the same number of templates to be created with various Collector sizes.

### Projects Used

To measure the impact of collector size on Auto-Learning duration depending on project size, 2 projects were used (1 collectable template; no index family / 2,000 templates no indexing). None of the projects had an Index Family. The goal was to measure the influence of the project initial size on Auto-Learning duration.

# ODBC Export

The ODBC Export testing scenario used a multipage image file containing 109 pages. Nine of those pages were patch codes and were discarded after being imported into ScanPlus. The 100 images processed by ODBC Export were 300 dpi binary.

THE SLEREXE COMPANY LIMITED

Phil.