Enterprise Content Management

Yariv Tzaban

Content Search Services

October 2014

Product Implementation Training (PIT)

Content Search Services (ECM Text Search) 5.2.1

Overview, What’s New, Some Advanced Topics

Introduction

- Course Overview
  - Content Search Services (CSS) overview, configuration, monitoring, and troubleshooting
- Target Audience
  - FileNet technical audience
- Suggested Prerequisites
  - FileNet background
- Version Release Date
  - October 2014

Course Objectives

After this course you will be able to:

- Describe what CSS is, its overall architecture, and folder structure
- Explain where to find useful logging and troubleshooting information
- Generate a CSS server status “ping page”
- Know where to start troubleshooting CSS

Content Search Services – ECM Text Search

- FileNet Content Search Services (CSS) is ECM Text Search (ECMTS)
- A scalable server component providing search capabilities via a client API
- Used as the CPE search component
- Used also by other enterprise products such as IBM FileNet CM, DB2 on many platforms, IBM Content Manager On Demand (CMOD) and IBM Content Manager (IBM CM)

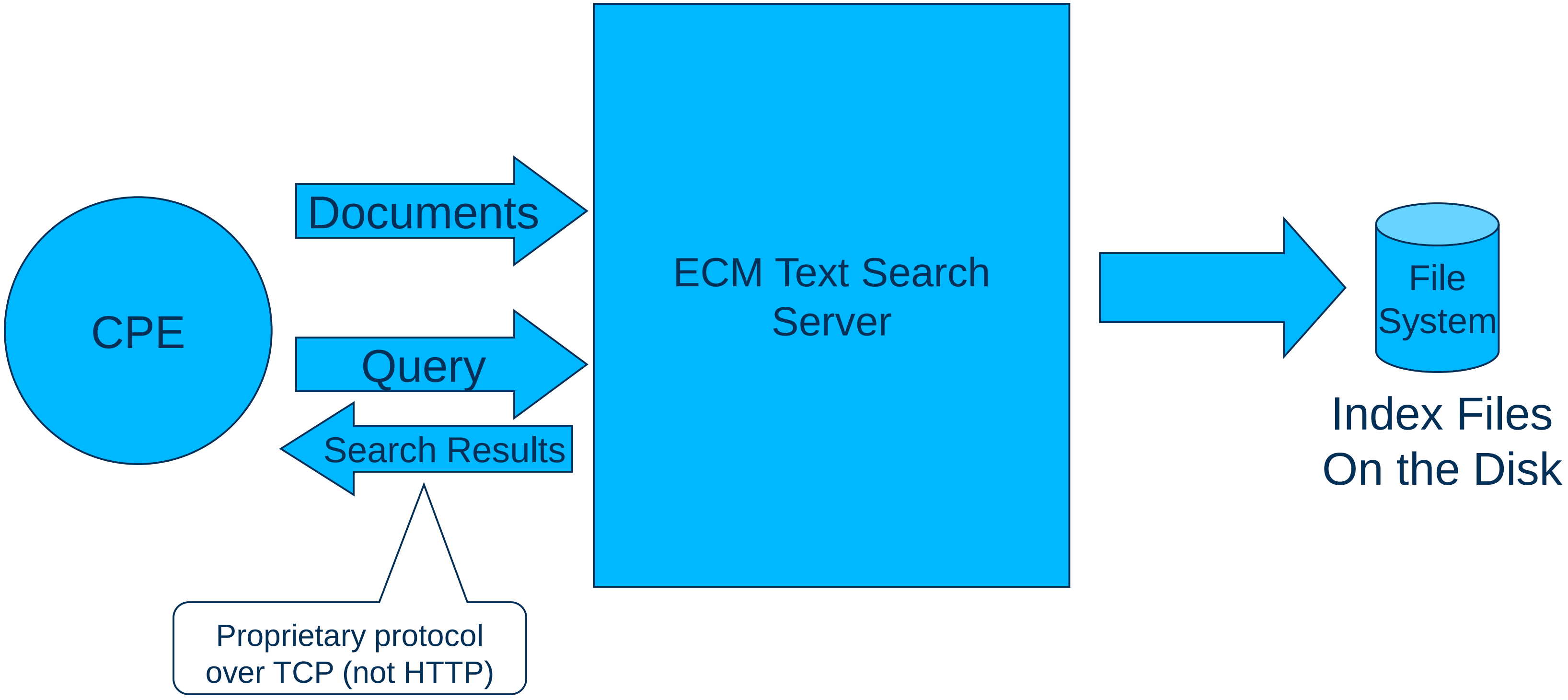
Content Search Services – ECM Text Search (2)

- Uses Apache Lucene (3.6.2) as the underlying search library for indexing and search, with enhancements for query processing, language processing, indexing & searching XML content, multilingual collections, etc.
- Full document processing pipeline including:
  - Text extraction from popular file formats (via Oracle Stellent OutsideIn library), archive files (TrueZIP, Junrar), etc.
  - Wide encoding support
  - Language identification
  - Tokenization and Language Processing in 22 languages

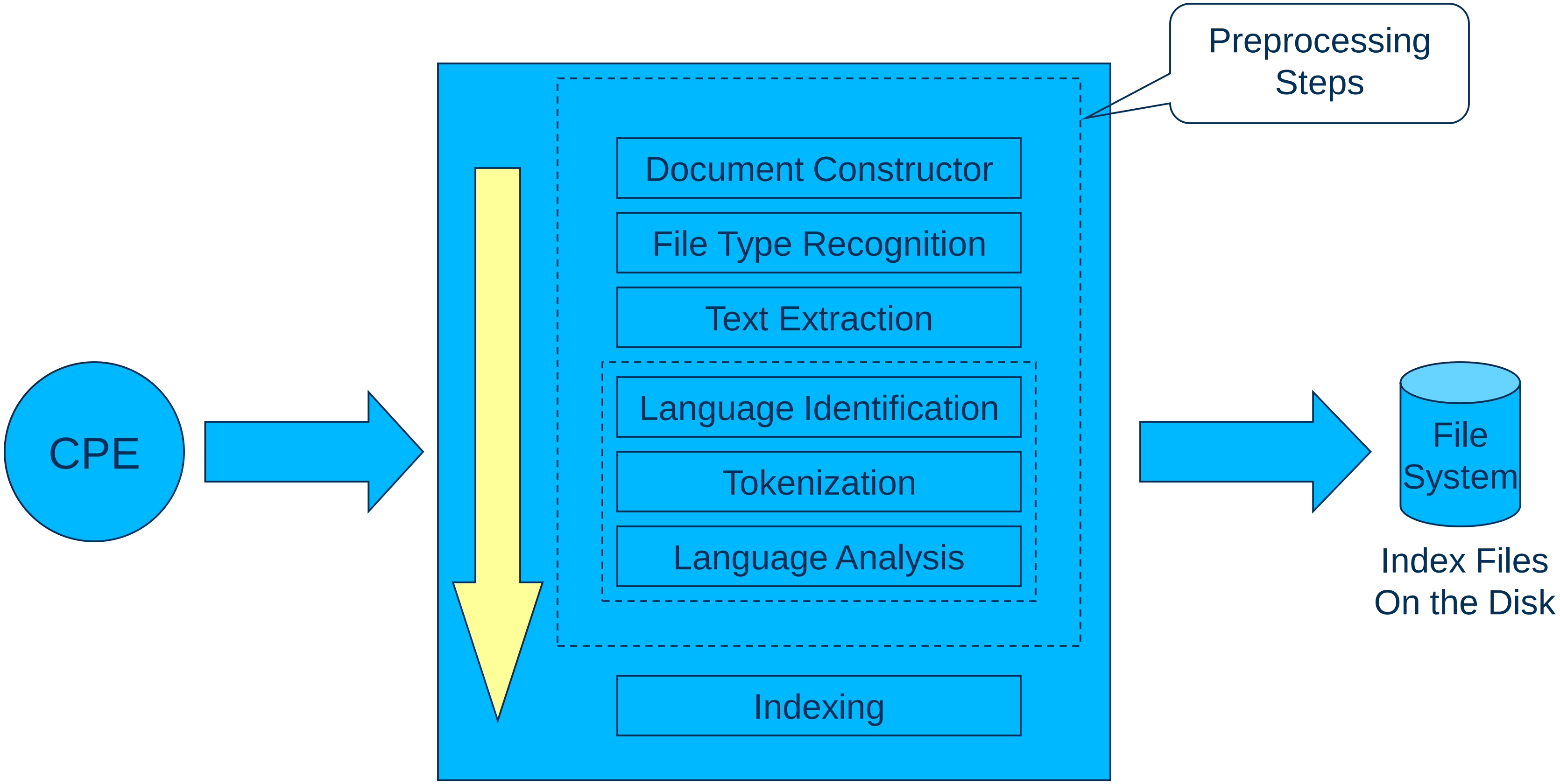
Architecture

CPE has 2 main interactions with the ECM Text Search server:

1. Indexing documents
2. Searching

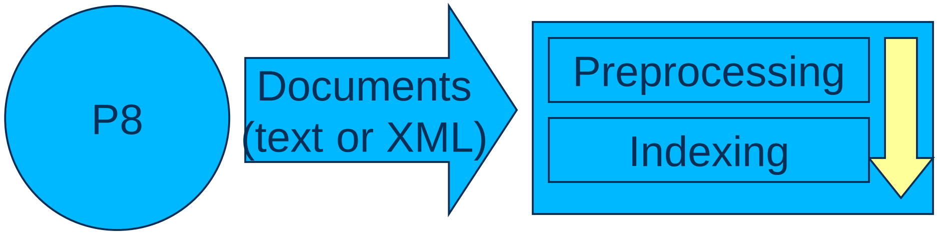


ECM Text Search – Indexing Pipeline



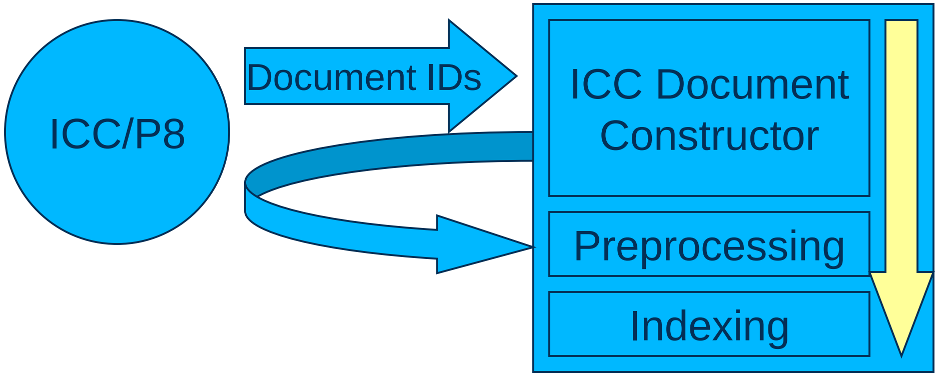
CPE standalone vs. ICC/CPE Indexing Architecture

1. When CPE sends documents for indexing, it sends text or XML documents (after extracting the text from binary documents)



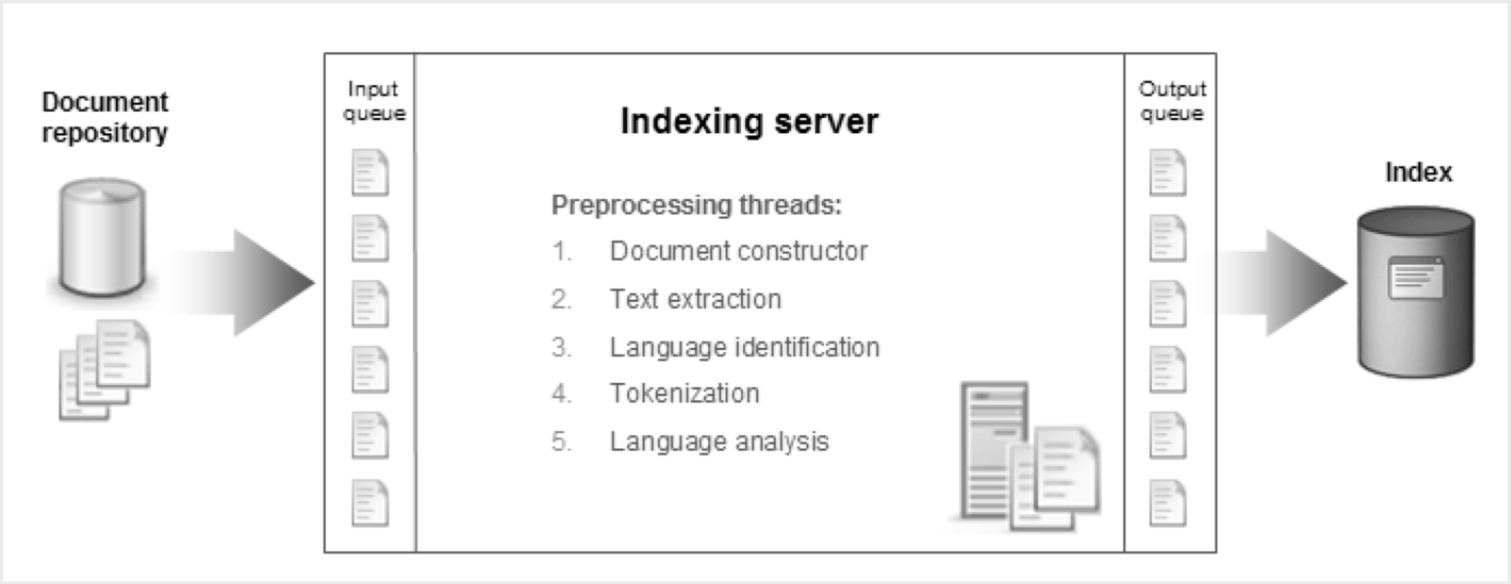
1. P8 Standalone Indexing

2. When ICC/CPE sends documents for indexing, it sends only the document ids. Then in the “ICC Document Constructor” (plug-in code running inside ECMTS server) it fetches the documents from the repository and extracts the text from binary documents.



2. ICC/P8 Indexing

Document Indexing Pipeline – Queues Architecture



ECM Text Search – Indexing Pipeline (2)

Document Constructor – Retrieves documents from the repository for further processing by ECMTS server *(An optional step. Used when ICC ingests documents into CPE, but not when CPE is used without ICC.)*

File Type Recognition – When the document content type or encoding is unknown, ECMTS can detect whether the document is binary or textual, and the document encoding

Text Extraction – Extracts textual content from document, based on file type and document encoding, including extracting content from archive files

ECM Text Search – Indexing Pipeline (3)

Language Identification – In order to apply language-specific analysis, the text's language is identified (if not given)

Tokenization – Text is broken into tokens using common delimiters (whitespace and punctuation) and language-specific techniques

Language Analysis – Tokens are analyzed using language-specific methods and dictionaries to identify lemmas (base forms of words), etc.

Indexing – Analyzed text submitted to the text index

CSS Platforms

- AIX POWER 6.1, 7.1 (64 bit)
- Linux SuSE ES 10, 11, Red Hat Enterprise Linux 6 (64 bit)
- Linux zSeries SuSE ES 10, 11, Red Hat Enterprise Linux 6 (64 bit)
- Solaris SPARC 10, 11 (64 bit)
- Windows Server 2008, 2012 (64 bit)

CSS Server Folder Structure

Default installation path on Windows: C:\Program Files\IBM\ECMTextSearch:

\bin – Batch/Shell scripts for starting/stopping, utilities, etc.

\lib – JAR files (including third party)

\config – Configuration files, default location of data files

\log – Generated log files

\resource – Resource bundles (for translation), and LanguageWare (dictionaries and configuration)

\stellent.<platform> (e.g., stellent.windows) – Text extraction library

\plugins – Constructor (plugins) JARs

\InstallerJVM – JVM used by installer/uninstaller

\Java60 – JVM used by product

\Uninstall_ECMTextSearch – Used by uninstaller

\MSRedist – (Windows only) Required Microsoft libraries

CSS Server Configuration Folder - \config

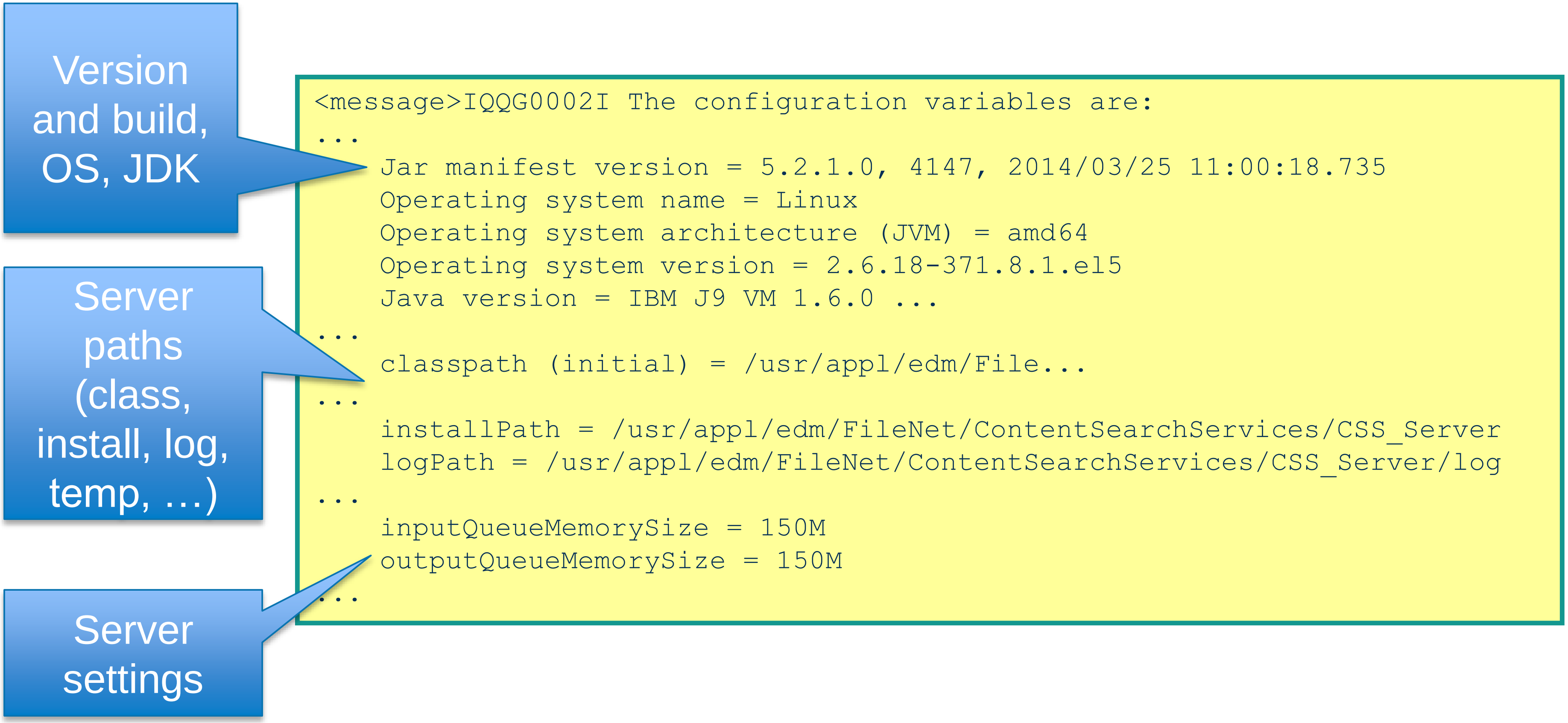
- config\configuration.xml – Many server settings (port, folders, queue sizes, timeouts, etc.)
- config\build_info.properties – Version information
- config\ecmts_logging.properties – Logging settings for server
- config\ecmts_config_logging.properties – Logging settings for utilities (configTool, adminTool)
- config\constructors.xml – Definition of constructors
- config\defaults\ – Default collection configuration file parser_config.xml
- config\defaults\binary_recognizer_config.xml – File type recognition settings
- config\collections\ – Default location of collections
- config\dictionaries\ – Stopword dictionaries

CSS Server Logging Folder - \log

- **trace0.log**, trace1.log, etc. – Primary log file, containing server logs according to configured logging level
- **monitor0.csv**, monitor1.csv, etc. – Server periodic diagnostic information in CSV format (ready for processing in Excel ;-)) such as number and size of processed documents (failed, succeeded), size of queues, concurrent queries, etc.
- **serverConfiguration.log** – Containing the current server configuration in effect, including JDK version, classpath, etc.
- **default0.log**, etc. – Server logs of logging level INFO and above
- **admin_audit0.csv**, etc. – Logs containing an audit trail of configuration changes
- **serverStatus_<date>_<time>.log** – Log containing output “ping” page after running “adminTool serverStatus” command
- **IndexingTrace/** – Folder containing indexing trace logs in case indexing trace was turned on (“configTool.bat/sh set –system –indexingTrace true”)

How do I find out the CSS server version and configuration?

At the beginning of trace0.log, and in serverConfiguration.log, you can see the



Version and build, OS, JDK

Server paths (class, install, log, temp, …)

Server settings

```
<message>IQQG0002I The configuration variables are:
...
    Jar manifest version = 5.2.1.0, 4147, 2014/03/25 11:00:18.735
    Operating system name = Linux
    Operating system architecture (JVM) = amd64
    Operating system version = 2.6.18-371.8.1.el5
    Java version = IBM J9 VM 1.6.0 ...
...
    classpath (initial) = /usr/appl/edm/File...
...
    installPath = /usr/appl/edm/FileNet/ContentSearchServices/CSS_Server
    logPath = /usr/appl/edm/FileNet/ContentSearchServices/CSS_Server/log
...
    inputQueueMemorySize = 150M
    outputQueueMemorySize = 150M
...
```

Server Status (“Ping Page”)

- New in CSS 5.2.1
- Running the server status command:

```
C:\Program Files\IBM\ECMTextSearch\bin>adminTool serverStatus -configPath ../config
The request was successfully executed.
The output was written to file: "c:\Program Files\IBM\ECMTextSearch\log\serverStatus_
20141021_103540.log".
```

- The server status is printed into the log folder (useful for collecting logs):

```
Report time = 2014-10-21T10:35:40
Server start time = 2014-10-21T10:35:35
Server uptime = 0 day(s), 0 hour(s), 0 minute(s), 5 second(s)

SERVER MEMORY STATUS
--------------------
Max Heap Size (-Xmx value) = 2048
Current Heap Size (MB) = 15
Used heap memory (MB) = 15
Unused heap memory (MB) = 0
Total number of open threads = 2
 SERVER MEMORY STATUS section took 0ms

ECMTS CONFIGURATION VARIABLES
-----------------------------
The configuration variables are:
...
```

Server Status (“Ping Page”) (2)

Things you can learn from the ping page:

- Server up time
- Server memory status
- Configuration settings in effect
- Document indexing queue status

  Same information you will find in monitor0.csv
- Open indexing tasks
- Query cache state
- Indexer cache state, searchable state cache
- Various threads state (indexing, preprocessing)
- Text extraction processes state
- Machine environment variables, JVM properties

Command-line Tools and Utilities

Various command-line tools can be found in the ECMTS_Home\bin directory:

adminTool

Manage collections, set trace options, and check the server version

configTool

Set system parameters and view system properties

dumpIndex

View the contents of a Lucene index

startup

A command line startup script

shutdown

A command line shutdown script

stopwordTool

Add or modify the list of stop words (frequently occurring terms that are removed from queries)

synonymTool

Add or remove synonym dictionaries from indexes

Viewing and Modifying Advanced Configuration Settings

- Viewing configuration settings at the system level:

```
C:\Program Files\IBM\ECMTextSearch\bin>configTool list –system [-details] [-showAdvanced]
```

- Viewing configuration settings at a collection level:

```
C:\Program Files\IBM\ECMTextSearch\bin>configTool list –collectionName <collection>
[-details]
```

- Modifying configuration settings:

```
C:\Program Files\IBM\ECMTextSearch\bin>configTool set –system [-parameterName] <value>

C:\Program Files\IBM\ECMTextSearch\bin>configTool set –collectionName <collection> [-
parameterName] <value>
```

ECMTS Configuration Model

There are 2 types of ECMTS configuration settings:

- Server
- Collection

ECMTS Configuration follows a simple hierarchical structure:

- Each setting has a default
  - Can be viewed using “configTool list”
  - Used in case the value for the setting is not set at the Server or Collection scopes
- Server settings can be set at the Server scope
- Collection settings can be set at the Server level, and at the Collection scope

```
Parameter name: securePort
Type: Integer
Current value: 55555
Default value: 0
Modifiable: true
Modifiable when server is running: false
Scope: system
Subsystem: server
Description: Specifies the number of the port on which the text search server will
listen to secure requests. You can disable the secure port by specifying a value of 0.
The default is 0.
```

Default Configuration Settings in CSS 5.2.1

Some configuration defaults were changed in CSS 5.2.1 to accommodate common environments:

- CSS server listener threads – 300 threads
- Max heap size – 3 GB
- Queue sizes – 150 MB
- Number of preprocessing and indexing threads – 8 threads

(Use the “configTool list” to view the defaults)

Troubleshooting Scenarios

The following are some problems that customers encountered and how they were approached:

- Error creating collection, error locking collection for indexing

  Possible causes:

  - Insufficient permissions on the collection folder
  - NFS mount options not sufficient/optimal for indexing (note about NFS reliability, hard mount vs soft mount, NFS stale directory problems, etc.)
  - Unstable communication

Troubleshooting – Is the Server Running?

Run Admin Tool command “version” or “status”

The “status” command lists the existing collections on the server, with their number of documents and size on the disk

In this example, I have a single collection “Base” with no documents:

```
C:\Program Files\IBM\ECMTextSearch\bin>adminTool version -configPath ../config
1.0.0.0-1.1-389

C:\Program Files\IBM\ECMTextSearch\bin>adminTool status -configPath ../config
CollectionName  IndexSize      NumOfDocuments
Base            16,604B        0
```

In this example, the server is not running:

```
C:\Program Files\IBM\ECMTextSearch\bin>adminTool status -configPath ../config
D0058E.CONNECT_FAILURE
IQQD0055E The search server is stopped. It must be started for the tool to run.
```

Troubleshooting – Out Of Memory

If an OutOfMemory error occurs, consider the following:

- Was an exceptionally large document indexed?
- Consider reducing maximal size of document
- Consider increasing size of max heap memory
- Consider reducing size of input and output queues
- Consider reducing number of server threads

Troubleshooting – Query not returning expected results

- Restart server (for IndexReaders to refresh)
- Turn on logging level to FINER
  - Review document parsing results in the log file
  - Review query parsing results in the log file
- Ensure document and query use the same language
  - Try providing an explicit language rather than relying on language identification
- Ensure document and query use the same collection
- Ensure query terms appear in the document
- Ensure query doesn’t contain stop words or synonyms that modify the query in an unanticipated way
- Simplify the query to validate smaller constraints first

Troubleshooting – Indexing Performance

Monitor Operating System Activity

- CPU
- Disk activity
- High memory consumption

Monitor ECMTS Queues Activity

- Input queue is empty
- Input queue is full, Output queue is empty
- Both queues are full
- Irregular queue activity

Review configuration and computer settings (gotchas)

- Resource limit requirements on Unixes
- Disk space
- Disk speed
- ECMTS threads (numberOfPreprocessingThreads, numberOfIndexerThreads)
- ECMTS queue sizes (inputQueueMemorySize, outputQueueMemorySize)
- ECMTS indexing parameters (MaxMergeMB, MergeFactor)

Contacts

- Development Manager: Yariv Tzaban
- Team Leader: Steve Kirshner
- Subject Matter Expert (SME): Shlomit Rosen
- Support: Wendy Taylor
- SME on the CPE side: Shawn Waters

Product Help/Documentation/Resources

Internal ECMTS 5.2.1 Documentation:

- ftp://pokgsa.ibm.com/projects/e/ecmts5.2.1/Docs/

CSS 5.2.1 Documentation:

- http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.relnotes.doc/wn_css.htm