



FAST INSTREAM

version 4.1

OPERATIONS GUIDE

Copyright

Copyright © 1997-2005 by Fast Search & Transfer, Inc. and its associated companies and licensors. All rights reserved. Fast Search & Transfer may hereinafter be referred to as FAST.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement. The software may be used only in accordance with the terms of the agreements. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's use, without the written permission of FAST.

Trademarks

FAST is a registered trademark of Fast Search & Transfer. All rights reserved.

FAST Search, FAST Data Search, and FAST InStream are trademarks of Fast Search & Transfer. All rights reserved.

Sun, Sun Microsystems, all SPARC trademarks, Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All rights reserved.

Netscape is a registered trademark of Netscape Communications Corporation in the United States and other countries.

Windows, Visual Basic, and Internet Explorer are registered trademarks of Microsoft Corporation.

Red Hat is a registered trademark of Red Hat, Inc. All rights reserved.

Linux is a registered trademark of Linus Torvalds. All rights reserved.

UNIX is a registered trademark of The Open Group. All rights reserved.

AIX is a registered trademark of International Business Machines Corporation. All rights reserved.

HP and the names of HP products referenced herein are either trademarks and/or service marks or registered trademarks and/or service marks of HP and/or its subsidiaries.

All other trademarks and copyrights referred to are the property of their respective owners.

Restricted Rights Legend

Software and accompanying documentation are provided to the U.S. government in a transaction subject to the Federal Acquisition Regulations with Restricted Rights. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

Contents

	FAST Support.....	vii
	About this Guide	ix
Chapter 1	Environment Variables	1
	Setting up Environment Variables on UNIX.....	2
	Setting up Environment Variables on Windows.....	2
Chapter 2	Starting and Stopping FAST InStream	3
	Starting FAST InStream	4
	Starting FAST InStream on a UNIX Node.....	4
	Starting FAST InStream on a Windows Node.....	4
	Shutting Down FAST InStream.....	5
	Types of Shutdown Procedures	5
	Running a Sequential Shutdown	5
	Procedure Overview.....	5
	Suspending the Connector Feed	6
	Suspending Document Processing	7
	Suspending Indexing on a Search Node.....	8
	Running a Simple Shutdown	9
	Running a Simple Shutdown on a UNIX Node.....	9
	Running a Simple Shutdown on a Windows Node.....	9
	Resuming Indexing on a Search Node.....	10
Chapter 3	Backing Up and Restoring FAST InStream	11
	Determining Backup and Restore Needs	12
	Basic Checklist.....	12
	Configuration	12

Content	13
Preparation for Backup	15
System Configuration Backup and Recovery	16
Backing up and restoring Content Connectors	17
Backing Up and Restoring Crawler Configuration and Data	18
Overview	18
Restore Crawler without restoring documents	18
Full Backup/Restore of Crawler Configuration and Data	18
Re-feeding the Cached Content of the Crawler	19
Forced re-crawling	20
Exporting and Importing Collection Specific Crawler Configuration	20
Backing Up and Restoring the Status Server	22
Overview	22
Status Server Recovery	22
Backing Up and Restoring the BMCP Database (Rank Tuning)	24
Backing Up the Rank Tuning Database	24
Restoring the Rank Tuning Database	24
Redeploying Rank Tuning Toolkit Information	24
Backing Up and Restoring Dictionaries	26
Preparing Backup and Restore	26
Backing Up	26
Restoring	26
Backing Up and Restoring the Anchor Server	28
Backing Up the Anchor Server Database	28
Restoring the Anchor Server Database	28
Adding New Anchor Text Boosts to already Processed Documents	28
Backing Up Index Data and Configuration	29
Introduction	29
Backing Up Index Data	29
Main Procedure	29
Optional Backup of Index Directory Data	30
Backing Up Index Configuration	30

Chapter 4	Index Data Failover/Recovery	31
	Deployment Scenarios for High Availability	32
	Introduction.....	32
	Reference Scenarios	33
	Failover/recovery Without Index High Availability.....	35
	Failover/recovery With Index High Availability.....	37
	Introduction.....	37
	Prerequisites	37
	Single Index/Search Node Failure.....	37
	Dedicated Search/Index Node	37
	Search/Index Node also including a Content Distributor	39
Chapter 5	Applying Run-Time Deployment Changes	41
	Capacity Scaling Dimensions	42
	Increasing Query Capacity and High Availability	43
	Adding a new Search Row	43
	Adding a Combined Index and Search Row	44
	Adding a Dedicated Search Row	46
	Adding a new Query and Result Engine	48
	Increasing the Content Volume Capacity	50
	Adding a Combined Index and Search Column	50
	Increasing Document Feed/Processing Capacity.....	54
	Performance Tuning of the FAST InStream Status Server.....	54
	Configuring new Search Nodes in the FAST Configuration Server	56
	Configuring a New Search Column in the FAST Configuration Server	56
	Configuring a New Search Row in the FAST Configuration Server.....	57
	Adding a Top-Level Dispatcher to a Single Search Node System	59
	Multi-node License File Considerations.....	62

Chapter 6	Installing Nodes from an Install Profile	63
	Install Profile Overview	64
	Node Installation Procedure.....	68
	Prerequisites.....	68
	Installing a Node According to the Install Profile.....	68
	Single Node Install on UNIX	68
	Single Node Install on Windows	69
	Installer Option Details.....	70
	Configuring new Nodes in the Install Profile	71
	Configuring a new Search/Index Node	71
Chapter 7	Adding a new Node using the OEM Installer	73
	Convert a single-node installation into a node in an existing multi-node installation	74
	Installing a Combined Search/indexer Node (new row or column)	74
	Installing a Dedicated Search Node	74
Chapter 8	Changing the Host Name of Your Installation	77
	Overview	77
	Changing the Host Name in a UNIX System	79
	Changing the Host Name in a Windows System	80
Chapter 9	Gathering Support Information	81
	Information Gathering	82
	Retrieving Product Version Information	82
	Retrieving Platform/Operating Systems Information.....	82
	Overall System Diagnosis Information.....	83
	Environment Variables.....	83
	System Environment	83
	Machine Environment.....	83
	License Status	84
	Machine Load and Memory	84
	Disk Space and CPU Usage.....	84
	IP/Network Addresses	85

Ports Allocation	85
Product Status Information	85
Process Status in Process Window	85
Documents Status	86
Log File Information	87

FAST Support

Website

Please visit us at:

<http://www.fastsearch.com/>

Contacting FAST

Fast Search & Transfer, Inc.
Cutler Lake Corporate Center
117 Kendrick Street, Suite 100
Needham, MA 02492 USA
Tel: +1 (781) 304-2400 (8:30am - 5:30pm EST)
Fax: +1 (781) 304-2410

Technical Support and Licensing Procedures

Technical Support for customers with active FAST Maintenance and Support agreements,

E-mail: tech-support@fastsearch.com

For obtaining FAST licenses or software, contact your FAST Account Manager or

E-mail: customerservice@fastsearch.com

For evaluations, contact your FAST Sales Representative or FAST Sales Engineer.

Product Training

E-mail: fastuniversity@fastsearch.com

Sales

E-mail: sales@fastsearch.com

About this Guide

Purpose of this Guide

The purpose of this guide is to provide necessary details to be able to analyze and prepare routines for backup and restore/recovery of a FAST InStream system, including moving configuration and/or data from one installation to another.

FAST InStream is a derivative of FAST Data Search for OEM deployments.

This guide is based on self-contained tasks. Nevertheless, it is recommended to read the guide entirely when planning the deployment of the system, to learn about what measures to take to ensure that the wanted level of recoverability is secured when the system is up.

Note! For information about other guides included in the FAST InStream documentation set refer to Chapter 1 *The FAST InStream Documentation Set* in the *Product Overview Guide*.

Audience

This guide provides information for several types of users:

- *Managers* and *supervisors*, who need to understand how FAST InStream functions in order to plan the system design, configuration, implementation, and operation.
- *System administrators* and *operators*, who need to understand the data flow and basic working in FAST InStream.

Conventions

This guide uses the following textual conventions:

- Terminal output, contents of plaintext ASCII files will be represented using the following format:

All hostnames should be fully qualified domain names.

What is the first node in the system?

- Terminal input from operators will be in the same but bold format:

What is the first node in the system? **host1.fast.no**

- Input of some logic meaning, will be enclosed in <> brackets:

Which port should the log server run on (default:16100)? **<enter>**

- URL's, directory paths, commands and the names of files, tags, and fields in paragraphs appear in the following format:

Run *Setup.exe* and follow the instructions.

- User Interface page/window texts, buttons, and lists appear in the following format:

Click on **Start Menu**.

- *\$FASTSEARCH* or *%FASTSEARCH%* refer to an environment variable set to the directory where FAST InStream is installed.
- File path syntax differences between Windows and UNIX (*\$FASTSEARCH/bin/* vs. *%FASTSEARCH%\bin*) are in most cases indicated explicitly. In some cases only the UNIX syntax is indicated for simplicity within the procedures. Unless otherwise stated the procedure is equal for Windows, except for the indicated file path syntax.

Chapter 1

Environment Variables

About this Chapter

The chapter provides the environment variables to use with FAST InStream.

It includes:

- Setting up Environment Variables on UNIX
- Setting up Environment Variables on Windows

Setting up Environment Variables on UNIX

To set up FAST InStream environment variables correctly, include the following command in the shell profile:

```
cd <path to FAST InStream installation>/bin; source ./setupenv.sh
```

If you are using a *bash* or *sh* related shells, add the command to the *.bashrc* or *.profile* file.

If you are using a *csh* related shell, you should normally add the command to the *.cshrc* file. For this shell the command is named `setupenv.csh`.

Note! For information about which variables FAST InStream uses, consult the appropriate script itself.

Setting up Environment Variables on Windows

The FAST InStream installer modifies the *PATH* variable and some other variables to support FAST InStream. You do not need to modify any variables on Windows before starting FAST InStream.

Chapter 2

Starting and Stopping FAST InStream

About this Chapter

This chapter contains information on how to start and stop FAST InStream and its components.

It includes:

- Starting FAST InStream
- Shutting Down FAST InStream
- Resuming Indexing on a Search Node

Starting FAST InStream

The following sections provide information on starting FAST InStream.

Note! You do not need to start FAST InStream explicitly after having installed it, as you can select whether to start FAST InStream or not during installation. The start procedures given here are only necessary if you chose not to start FAST InStream right after installation or if it was stopped.

Starting FAST InStream on a UNIX Node

Start FAST InStream system on UNIX by using the following command:

```
$FASTSEARCH/bin/nctrl start
```

This will start all the processes currently marked as active. Ensure that environment variables are set properly first, refer to Chapter 1 *Environment Variables*.

In a multi-node installation this procedure must be performed on all the nodes, starting with the FAST InStream administration node.

Note! Note that no measures are taken by the installer to ensure that FAST InStream is restarted after a system reboot.

Starting FAST InStream on a Windows Node

On Windows systems, FAST InStream runs as a Service per default. Per default, it is also accessible from the **Start** menu. To start FAST InStream from the taskbar, select:

Start → Program → Fast Data Search → Fast Data Search -> Start

Alternatively, you may use the following net command:

```
net start FastDSService
```

In a multi-node installation this procedure must be performed on all the nodes, starting with the FAST InStream administration node.

Shutting Down FAST InStream

Types of Shutdown Procedures

There are 2 types of shutdown procedures in FAST InStream:

- a sequential shutdown (recommended)

A sequential shutdown involves shutting down all components individually and in a specific sequence. This is necessary in order for all components to keep the same view of the content after a restart and to maintain consistent behavior and data before and after the shutdown.

- a simple shutdown

A simple shutdown consists of shutting down a node completely without having to shut down individual components first. This might be helpful in a testing environment where consistency of data is not required, or in a system where documents are regularly re-fed and will eventually be re-entered and reprocessed at a later time. In such cases, the possibility of a few documents not being available for a limited period of time may not be as critical.

Note! Regardless of which type of shutdown you choose, you need to run it on all nodes in your system, as the command-line tool `nctrl` that is used for shutdown works on a per node basis only.

Running a Sequential Shutdown

Procedure Overview

To run a sequential shutdown, complete the following procedure:

- 1 Suspend all active Connectors (see section *Suspending the Connector Feed* on page 6) on any node.
- 2 Suspend ongoing document processing (see section *Suspending Document Processing* on page 7) on any node.
- 3 Shut down all FAST InStream processes involved in document processing; this includes all nodes that are running document processors or content distributors.
- 4 Shut down all Query and Result Engine servers (qrserver processes) on any node.
- 5 Suspend indexing (see section *Suspending Indexing on a Search Node* on page 8) on all indexer nodes.
- 6 Shut down indexer and search nodes on any nodes.

- 7 Make sure the Status Server is shut down after indexers to avoid losing callbacks on processed documents. Alternatively, suspend indexing on all indexers (see section *Suspending Indexing on a Search Node* on page 8). Note that a suspended index will remain suspended after a stop/start sequence, so the indexer must be explicitly resumed later on.
- 8 Shut down the Administration services by stopping FAST InStream on the remaining nodes.

Suspending the Connector Feed

This section describes how to stop the Crawler and the File Traverser. If there are any other FAST Connectors running within your FAST InStream installation, they need to be suspended using the procedures provided in the respective FAST Connector documentation.

Suspending the FAST Crawler

- In a single node environment, you may suspend the FAST Crawler by either:
 - selecting the **stop** symbol for the Crawler on the **System Management** screen in the administrator interface
 - running the following command:

```
> nctrl stop crawler
```
- In a multi node environment, you suspend the FAST Crawler by running the following command:

```
> adminclient -x
```

on the node running the Crawler ubermaster.

The crawler will not stop completely before all outstanding content batches have been successfully submitted to FAST InStream and received by the indexer nodes. You may monitor the crawler submit queue by waiting until the following folder (on the node running the crawler) is empty:

<code>\$FASTSEARCH/data/crawler/dsqueues</code>	(UNIX)
<code>%FASTSEARCH%\data\crawler\dsqueues</code>	(Windows)

Suspending the FAST Filetraverser

The FAST File Traverser does not support an explicit suspending or resuming operation. Either a run must continue to completion, or if you choose to interrupt a file traverse being executed, you need to re-feed a run from the start again upon system restart.

Suspending any other FAST Connector

Refer to the respective FAST Connector documentation for information on how to stop the specific FAST Connector.

Note! After the Connectors have been stopped, there may still be documents in the pipeline within the system. To ensure that no more content is being processed, you need to stop content processing as well (see section *Suspending Document Processing*).

Suspending Document Processing

Note! If all Connectors in use keep track of the status of individual updates, and are able to re-feed not completed documents after restart, then this step can be safely skipped.

To suspend document processing by selecting the suspend symbol for all document processors on the **System Management** screen in the administrator interface.

The callback mechanism informs a connector or API client about the status of outstanding content batches. Once the batch is received and persisted on disk by the Indexer there will be a `persisted_by_indexing_engine` callback to the connector (refer to the *Content Integration Guide* for details). This callback guarantees that the documents will eventually be indexed. This guarantee applies even if the system is restarted. Documents whose batch has not received this callback, may not be persisted before the system terminates.

The information about the state of individual documents is present in the Status Server, but is currently not available in the external APIs. If this information is needed, contact FAST Professional Services at FAST Support.

Note! In general, the safest strategy for a Connector is to re-feed all documents for which the final callback has not been received when the system comes back up.

Suspending Indexing on a Search Node

After a successful stop of both content feeding and document processing, no more documents will enter the indexer queues. However documents may already have been queued to being indexed. To ensure that no more documents are being indexed, indexing must also be suspended.

To flush document queues and suspend indexing on a search node, run the following command on the node:

```
rtsadmin <nameserverhost> <nameserverport> <clustername> <columnid>  
<rowid> suspend
```

where:

- <nameserverhost> is the name of the host running the nameserver. Refer to *\$FASTSEARCH/etc/omniorb.cfg* and the line starting with "InitRef"
- <nameserverport> is the port of the Name Server. Refer to *\$FASTSEARCH/etc/omniorb.cfg* and the line that starts with "InitRef"
- <clustername> is the name of the current search cluster (default: webcluster)
- <columnid> is the index column ID, starting with 0 (zero)
- <rowid> is the search row ID, starting with 0 (zero)

The row and column ID can be found from the **Matching Engines** screen in the administrator interface. Select the appropriate Search Column and select **Status Details**. The row/column ID is displayed.

Note! Repeat this command for each node that runs an indexer.

Suspending indexing is useful to ensure a consistent and known state of the index on disk before backing it up, or during shutdown and restart of whole or parts of the system.

Running a Simple Shutdown

Stopping FAST InStream with a simple shutdown means the immediate stop of all FAST InStream processes. This procedure does not ensure consistent behavior or data after a restart of the system and is therefore recommended only for test environments or installations in which momentarily missing data is not critical.

Running a Simple Shutdown on a UNIX Node

Stop all FAST InStream processes on a UNIX node by running the following command:

```
$FASTSEARCH/bin/nctrl stop
```

Running a Simple Shutdown on a Windows Node

Stop all FAST InStream processes on a Windows node from the taskbar by selecting:

Start → Program → Fast Data Search → Fast Data Search → Stop

Alternatively, you may use the following net command:

```
net stop FastDSService
```

Resuming Indexing on a Search Node

To resume indexing after suspension, issue the following command to restart indexing:

```
rtsadmin <nameserverhost> <nameserverport> <clustername> <columnid>  
<rowid> resume
```

where the parameters are described in section *Suspending Indexing on a Search Node* on page 8.

This action re-enables the content input queues on the search node.

Note! Repeat this command for each node that runs an indexer.

Backing Up and Restoring FAST InStream

About this Chapter

This chapter provides considerations and procedures on what, when, and how to restore critical data in your FAST InStream installation.

It includes:

- Determining Backup and Restore Needs
- Preparation for Backup
- System Configuration Backup and Recovery
- Backing up and restoring Content Connectors
- Backing Up and Restoring Crawler Configuration and Data
- Backing Up and Restoring the Status Server
- Backing Up and Restoring the BMCP Database (Rank Tuning)
- Backing Up and Restoring Dictionaries
- Backing Up and Restoring the Anchor Server
- Backing Up Index Data and Configuration

Determining Backup and Restore Needs

Basic Checklist

Decisions on what parts of a FAST InStream system to backup are closely related to decisions made during deployment. Answer the following questions to help you determine which scenarios your particular installation needs:

- Is all the content available to be re-fed upon need?
- Is it acceptable to perform a full reprocessing and indexing overhead of all nodes if one indexer node fails?
- Is it necessary to separately handle an indexer failure, or will it be sufficient to restart/reinstall the index partition with empty indexes and let all content be updated to eventually arrive at a complete index again?

There are two types of information that you need to backup and restore: *Configuration* and *Content*.

Configuration

Configuration of FAST InStream can be divided into the following subcategories:

- Install Configuration (one per FAST InStream system):

The basic layout of the system, distribution of tasks and mapping to nodes, are created by the FAST InStream installer, and saved in *\$FASTSEARCH/InstallProfile.xml*. Upon reinstallation of a system or installation of a copy of a system, select advanced mode and supply the saved configuration file kept from the earlier install. If desired, it is possible to run the installer to create a new index profile without starting the installation.

How to re-install a FAST InStream node from the Install Profile is described in Chapter 6 *Installing Nodes from an Install Profile*.

- Index Profile (one per cluster)

The index profile is the toplevel interface to how documents are represented and stored on disk, and to what features the installation makes available. Details of the index profile features are documented in the *FAST InStream Configuration Guide*.

- Collection Specific Configuration

Collection specific configuration information is in use in several components:

- Certain Data Sources maintain their own collection specific information. In particular, most of the configuration information associated with the crawler is in the form of a collection specific configuration, such as what URLs to crawl or document types to accept. See section *Backing Up and Restoring Crawler*

Configuration and Data on page 18 for details on how to backup and restore crawler related configuration and data. You can also export/import configuration per collection, see section *Exporting and Importing Collection Specific Crawler Configuration* on page 20. For other connectors, consult the relevant connector guide for details.

- The Configuration Server maintains a list of all existing collections in *\$FASTSEARCH/etc/CSConfig.xml*. This file is updated by the Configuration Server upon exit. See section *System Configuration Backup and Recovery* on page 16 for details.
- A running Status Server database keeps a cache of valid collection names, read from Configuration Server at startup, and maintained by internal APIs upon creation and deletion of collections. See section *Backing Up and Restoring the Status Server* on page 22 for details
- Document Processing Configuration

Document processing configuration consists of configuring pipelines, processing stages, and dictionary configuration setup. Pipeline and processing stage configuration is maintained by the Configuration Server, see section *System Configuration Backup and Recovery* on page 16.

If custom dictionaries have been made, these must also be backed up, both source and compiled versions (.aut files), in addition to any custom document processor stages. Refer to section *Backing Up and Restoring Dictionaries* on page 26.

Content

In this context we consider Content as the different representations of the submitted/indexed data within FAST InStream. This includes:

- Content related repositories within FAST InStream. From a backup/restore perspective the following content related repositories must be considered:
 - *Data Sources*. Refer to section *Backing up and restoring Content Connectors* on page 17.
 - The *Status Server* stores state and routing information, such as what indexer column holds a particular document, for all documents in the system. Refer to section *Backing Up and Restoring the Status Server* on page 22.
 - *Index Data*. Refer to section *Backing Up Index Data and Configuration* on page 29.
 - *Document Processing* related repositories.

If *BMCP Rank Tuning* or the Rank Tuning Bulk Loader is used, the associated database must be backed up. Refer to section *Backing Up and Restoring the BMCP Database (Rank Tuning)* on page 24.

If *Authority Ranking* (based on anchor text) is in use for WEB content (by default in the *SiteSearch* and *NewsSearch* pipelines), the associated database must be backed up. Refer to section *Backing Up and Restoring the Anchor Server* on page 28.

- Content being passed through the system. At the time of a backup the system should be placed in a stable state where content feed is suspended.

Which parts of the content flow that is backed up is mainly related to how long it will take to restore a FAST InStream system to the state it was before a system failure.

- A full re-feed and re-index may be acceptable if search down-time equal to the time it takes to re-feed and re-index is acceptable. This is the simplest scenario. In this case you only need to backup the system configuration.
- For certain applications such as using the crawler a full re-feed takes much longer time than re-indexing. For such applications it may be acceptable to re-index but not re-feed. Content fed into FAST InStream using the content APIs or connectors will first pass through document processing, which normally does not involve any storing of content to disk. Then documents will be fed to the indexer, which will store it to disk using the FAST internal indexer XML format called *fixml*. By backing up the *fixml* data only it is possible to recover in reasonable time, where each indexer will perform a full re-index from the *fixml*.
- For high-availability applications it may also be considered to backup the entire index data.

Refer to section *Backing Up Index Data and Configuration* on page 29 for further details.

It is important to understand that only the data that needs to be available for indexing and result processing/presentation needs to be represented in the *fixml*, FAST InStream does not store the original content as such, except for the Enterprise Crawler (see below).

Normally, to be able to reprocess documents after changes in the document processing setup, one of the following assumptions must hold:

- Original content can be re-fed on demand – no need to store it in FAST InStream in its original form (file traverser assumes this, and does not attempt to store the documents).
- The connectors maintain storage of the fetched documents (this is true for the crawler in the default configuration). For the crawler this is only in order to be able to re-feed original content due to document processing changes, not in order to be able to retrieve the original document by a search application.

Preparation for Backup

Before performing a full backup you should suspend feeding and indexing in order to ensure a consistent state in your FAST InStream system:

- 1 Stop feeding from the content source as described in section *Suspending the Connector Feed* on page 6.
- 2 Give the document processing some time to finish outstanding batches submitted by the connector(s). When using the Enterprise Crawler you can monitor this using the procedure as described in section *Suspending the Connector Feed* on page 6.
- 3 Suspend indexing on all working indexing nodes within the column as described in section *Suspending Indexing on a Search Node* on page 8.

System Configuration Backup and Recovery

The following files contains important configuration server data that should be backed up from the Administration node in your FAST InStream system:

```
$FASTSEARCH/InstallProfile.xml  
$FASTSEARCH/etc/CSConfig.xml  
$FASTSEARCH/etc/CSUniCfg.xml  
$FASTSEARCH/etc/PipelineConfig.xml  
$FASTSEARCH/etc/DefaultCrawlerConfig.xml
```

In addition you need to maintain a copy of your Index Profile for backup.

The following scenario describes a situation where the Admin node (including the Configuration Server and Status Server) has crashed and needs to be reconstructed:

- 1 If the disk system is operative, you may back up the system that has failed to whatever extent is possible. Otherwise the latest backup of the system must be used.
- 2 Resolve the hardware problem that caused the failure.
- 3 Re-install the node from the install profile according to the procedure described in section *Node Installation Procedure* on page 68. This will re-install but not start the services on the recovered node.
- 4 Restore the Configuration Server files (*CSConfig.xml*, *CSUniCfg.xml*, *PipelineConfig.xml* and *DefaultCrawlerConfig.xml*) from backup.
- 5 Restore other data from the latest backup. Refer to the other sections in this chapter.
- 6 Start FAST InStream on the recovered node. Refer to section *Starting FAST InStream* on page 4 for details.

Backing up and restoring Content Connectors

The Content Connectors may or may not store content or references to content, depending on how the connectors are set up. There are four main types of connectors or connector configurations:

- 1 A simple batch retrieve connector will retrieve all or a subset of documents from a content repository and submit the documents unconditionally to FAST InStream. Such a simple connector will not involve any content store within the connector, and hence no need for backing up content.
- 2 Certain connectors use a database to store change control information, for instance represented as a hash value for the document. By using this hash database it is possible to determine whether a retrieved document is changed or not, and hence only forward changed documents to FAST InStream.

Backing up the change control database is optional, and will only avoid the need for re-submitting all documents to FAST InStream after a recovery. The need for such a backup depends on how long it will take to re-fetch all content from the content source.

- 3 Some connectors subscribe to change information that is available from the content source, and does only retrieve new and changed documents. In this case the change information does not need to be stored in the connector, and there will normally not be any need for backing up content information.
- 4 The Enterprise Crawler also stores the original web documents in order to support re-feed of documents upon document processing changes. This includes pipeline changes and changed BMCP Rank Tuning data or new Anchor Texts detected.

The FAST Content Connectors may have been configured according to (1), (2) or (3). Consult the documentation for each connector for details.

The FAST File Traverser is normally used according to (1). Refer to the *File Traverser Guide* for details.

Backing Up and Restoring Crawler Configuration and Data

Overview

Crawler configuration is primarily concerned with collection specific settings. Backup of the crawler configuration will ensure that the crawler can be reconstructed to a state with identical setup, but without knowledge of any documents. The crawler configuration is located in `$FASTSEARCH/data/crawler/config/config.db`. To backup the configuration, save this file.

It is also possible to export/import collection specific crawler configuration using the Enterpriser Crawler *adminclient* tool. Refer to section *Exporting and Importing Collection Specific Crawler Configuration* on page 20. However, this is not necessary for pure backup needs (the config.db file includes all the collection specific information). The *adminclient* can be used for simple monitoring tasks related to the crawler. FAST Professional Services can provide more details upon request.

Restore Crawler without restoring documents

To restore a node to the backed up configuration without restoring the documents:

- 1 Install the node according to the procedure in section *Node Installation Procedure* on page 68.
- 2 Ensure that the crawler is not running, then restore the backed up config database `$FASTSEARCH/data/crawler/config/config.db`.
- 3 Restart the crawler. It will then start re-crawling from the start according to the restored configuration.

Full Backup/Restore of Crawler Configuration and Data

Backing up the crawler configuration only ensures that all information about individual collections and the setup of the crawler itself can be restored, but will trigger the sometimes unacceptable overhead of having to crawl and reprocess all documents over again. To be able to recover without this overhead, a full backup of the crawler is needed.

Backing up

- 1 Stop the crawler according to the procedure in section *Suspending the Connector Feed* on page 6.
- 2 Backup the complete `$FASTSEARCH/data/crawler` directory on all nodes involved in crawling.

- 3 If keeping log files is desired, backup the *\$FASTSEARCH/var/log/crawler* log file directory. This is for reference only, and is not needed to get the system backup.

Restoring

- 1 Install the node according to the procedure in section *Node Installation Procedure* on page 68.
- 2 Make sure the crawler is not running, then restore the backed up *\$FASTSEARCH/data/crawler* directory on each node to be restored.
- 3 Restart the crawler. It will then start re-crawling from the point where it was backed up, and according to the restored configuration.

Re-feeding the Cached Content of the Crawler

This section describes how to re-feed the currently cached content of the crawler into the document processing pipeline without starting a re-crawl.

If you have a configuration scenario with the crawler on a separate node you may experience a recovery situation where index data has been lost (when running single-row indexer). If the crawler node is fully operative, it is recommended to perform a full re-feed of crawled documents. This can be time-consuming, but may be the only way to ensure full recovery of documents submitted after the last backup.

Note! This procedure is for the single node crawler case only; for multinode crawler setup contact FAST Professional Services.

- 1 Stop the crawler according to procedure in section *Suspending the Connector Feed* on page 6.
- 2 Run the **postprocess** program in manual mode, using the **-R** (re-feed) option:
 - a To re-feed a single collection, use:

```
> $FASTSEARCH/bin/postprocess -I master -R <collectionname>
```
 - b To re-feed all collections, use:

```
> $FASTSEARCH/bin/postprocess -I master -R ""
```
- 3 When **postprocess** has completed the feeding, it displays **Press Ctrl-C to abort**. At this point it is safe to terminate it, since **postprocess** has identified and enqueued all documents due for processing, as long as the crawler is later restarted so that the feeding of the remaining documents can be serviced. The remaining documents will then eventually be processed before any newly crawled data. Otherwise, **postprocess** will eventually terminate itself when all documents has been processed.

- 4 Finally, restart the crawler again in normal mode:

```
> $FASTSEARCH/bin/nctrl start crawler
```

Forced re-crawling

The procedure in section *Re-feeding the Cached Content of the Crawler* on page 19 assumes that the crawler database is correct. This implies that it will only re-feed all already crawled documents to FAST InStream. In case of a single-node failure or crawler node failure the last documents fetched by the crawler (after last backup) will be lost. In this case you must instead perform a full re-crawl of all the collections. This will then re-fetch the remaining documents. The following command will force a full re-crawl of a given collection:

```
> adminclient -F <collection>
```

Repeat this command for each collection in the system. This will then ensure that all documents crawled after the last backup will be re-fetched. Note that the re-crawl may take a reasonable amount of time before finished, but the index will be fully operative in the meantime.

Exporting and Importing Collection Specific Crawler Configuration

The basic collection data is backed up (exported) and restored (imported) using the procedure described in section *System Configuration Backup and Recovery* on page 16. This, however, does not include the data source configuration for the Enterprise Crawler.

The Enterprise Crawler configuration can be set and read from the Administration GUI, but it is also possible to export/import crawler configuration using the command-line utility *\$FASTSEARCH/bin/adminclient*. This is a convenient tool for exporting the Enterprise Crawler setup of a particular collection to a human readable form (XML) which also can be imported/restored. Some useful commands are:

Save collection configuration to a file:

```
adminclient -G <collectionname> > collection.xml
```

Restore one or more collections from a file:

```
adminclient -f collection.xml
```

Remove a collection from the crawler's knowledge:

```
adminclient -d <collectionname>
```


Important multinode crawler note: In a multinode crawler, the *adminclient* application must always be run on the main crawler node (the node running the ubermaster) to ensure that all nodes are updated.

Note! Use the `-h` command for help on other *adminclient* operations.

Backing Up and Restoring the Status Server

Overview

The Status Server stores state and routing information, such as what indexer column holds a particular document, for all documents in the system. This information is persisted under *\$FASTSEARCH/data/status*. If this directory structure is backed up, the Status Server can be easily restored to its previous state.

The Status Server recovers its data structure completely from the information available in the indexer nodes. This recovery procedure can be triggered by the user, but will automatically be initiated whenever the Status Server discovers that it is not in sync with the indexers. In FAST InStream the recovery strategy is simple – the complete state of the Status Server is built up from scratch under the assumption that the indexers have the right information.

Consequently, a backup of the Status Server state is only valuable if feeding has been stopped. If there is a chance that indexer information is out of sync with the information in the Status Server and if available, its backup, the only way to get to a consistent state is by triggering a Status Server recovery. Right after a recovery procedure, the Status Server will know about the documents that are available in the indexers, but will have lost information about all documents that had previously been fed, but that for some reason were lost in the indexers, for instance as a result of a node crash in one column.

Status Server Recovery

The Status Server supports user activated recovery which will trigger a rebuild of its complete state from the indexers. Normally, this should not be necessary, since the Status Server and indexers will be kept in sync automatically. However, there are situations where manual changes have been applied to the indexers, and where such changes need to be reflected into the Status Server structures to avoid duplicates.

The Status Server will rebuild its state with the following procedure:

For **Windows**:

Select **statusserver.exe** in Task Manager. Right click and select **End Process**. Then enter the FAST InStream administrator interface and start the **Status Server**. Upon restart, the Status Server will do a recovery because it was not properly shut down.

The Windows strategy can also be used on all supported UNIX platforms by killing the application with signal **-9** to force an unclean shutdown.

Note that the Status Server will automatically suspend feeding and flush all processor server pipelines before starting up recovery, to ensure that no updates arrive at the search nodes while the Status Server and the search nodes are reconstructing the Status Server's system view. Once recovery is complete, the Status Server will re-enable feeding.

For **UNIX (Linux)**:

```
killall -HUP statusserver
```

For **UNIX (Solaris, AIX, HP-UX)**:

```
kill -HUP `cat $FASTSEARCH/data/status/statusserver.pid`
```

Backing Up and Restoring the BMCP Database (Rank Tuning)

The *BMCP Rank Tuning* module and the *Rank Tuning Bulk Loader* maintains a run-time database used by the rank boost processing stage. You should back up this database to ensure that manual rank boost data is preserved.

Backing Up the Rank Tuning Database

- 1 Stop the storage service:
`$FASTSEARCH/bin/nctrl stop storageservice`
- 2 Back up the directory `$FASTSEARCH/data/rdbms`.
- 3 Restart the storage service:
`$FASTSEARCH/bin/nctrl start storageservice`

Restoring the Rank Tuning Database

- 1 Stop the storage service:
`$FASTSEARCH/bin/nctrl stop storageservice`
- 2 Restore the directory `$FASTSEARCH/data/rdbms`.
- 3 Restart the storage service:
`$FASTSEARCH/bin/nctrl start storageservice`

Redeploying Rank Tuning Toolkit Information

Normally, rank-tuning information is applied to new documents during processing and indexing. However, in some cases (after fatal recovery situations) it may be that the only option is to restore content from a backup taken before the latest rank tuning changes. In such cases restored documents in the index may not have the newest rank tuning information, in which case this procedure is necessary.

- 1 Ensure that the storage service is running.
- 2 Start a MySQL client:
`$FASTSEARCH/rdbms/bin/mysql`
`--defaults-file=$FASTSEARCH/rdbms/datasearch.cnf`
`-u root`

- 3 Delete the last deployment dates in the vespa database and the data manager:

```
% use vespa;  
% delete from RankTuningDeployRecord;  
% use datamanager;  
% UPDATE FROM dm_table_state SET deploy_started=0, deploy_finished=0  
% exit
```

- 4 Stop the storage service:

```
$FASTSEARCH/bin/nctrl stop storageservice
```

- 5 Start the cache manager:

```
$FASTSEARCH/bin/nctrl start cachemanager
```

- 6 Flush the cache manager's caches:

```
$FASTSEARCH/bin/cmctrl flush ranktune_doc_querybo  
$FASTSEARCH/bin/cmctrl flush ranktune_doc_staticboost
```

- 7 Restart the storage service:

```
$FASTSEARCH/bin/nctrl start storageservice
```

- 8 In the Business Manager's Control Panel (BMCP), press the deploy button for all collections in the system.

Backing Up and Restoring Dictionaries

Dictionaries marked as "open" in the *dictman* client only reside in memory, and must be saved to make changes permanent. Refer to Chapter *Configuring Linguistics Processing* in the *Configuration Guide* for details on using the *dictman* tool.

Preparing Backup and Restore

- 1 Ensure that no other users are modifying dictionaries.
- 2 Enter the dictman tool:

```
$FASTSEARCH/bin/dictman -u root
```

For reference, the application server user database is persisted in vespa mysql-database in the tables User, Roles and RolesUsers.
- 3 Issue the command list to determine if any dictionaries are open.
- 4 Close any open dictionaries by repeated use of the command close *<dictionary>* for each of the open dictionaries to save.

Backing Up

- 1 Ensure that all dictionaries that have been modified are saved. Stop the j2ee server, either from the Administration Interface or using the following command:

```
$FASTSEARCH/bin/nctrl stop j2ee
```
- 2 Backup master dictionaries by saving the content of *\$FASTSEARCH/j2ee/resources/dictionaries* on the node running the j2ee server.
- 3 Compiled dictionaries (automata) are located in *\$FASTSEARCH/resources/dictionaries* on each node. These automata can be reconstructed from the master dictionaries as long as their type is supported by dictman (spell check, lemmatization, synonym, and phrase dictionaries are supported) and corresponding master dictionaries are available. Note that such reconstruction may take a while to complete, so it is a good idea to backup the compiled versions as well. Other directories (such as "stopwords" and "util") have to be backed up.
- 4 Restart the j2ee server, either from the Administration Interface or using the following command:

```
$FASTSEARCH/bin/nctrl start j2ee
```

Restoring

- 1 Make sure that the j2ee server is not running:

```
$FASTSEARCH/bin/nctrl stop j2ee
```
- 2 Restore master dictionaries to the node running the j2ee server by restoring the files to

\$FASTSEARCH/j2ee/resources/dictionaries.

- 3 Restore any compiled dictionaries that were backed up to *\$FASTSEARCH/resources/dictionaries.*
- 4 Start dictman and enter the command **compile**. You will get a job number back.
- 5 Use **jobstatus <nr>** to verify that compilation has finished (status turns to "ok").
- 6 From the shell, copy the resulting files in *\$FASTSEARCH/j2ee/work/compile/dictionaries* to *\$FASTSEARCH/resources/dictionaries* on all nodes that contain a processor server or a query and result processor.
- 7 Restart the j2ee server:

\$FASTSEARCH/bin/nctrl start j2ee

Backing Up and Restoring the Anchor Server

The Anchor Server provides the document processing part of the Authority Boost feature. This feature enables adding citation information (anchor texts in web documents) from other documents to the metadata of the given document. This process implies that a document must be re-indexed every time new anchor texts pointing to this document has been found. Refer to Chapter *Index Profile Feature Management* in the *Configuration Guide* for further details on configuring Authority Boost.

The Anchor Server may have been configured to run on multiple nodes. In that case the following procedures must be applied to all nodes running an anchorserver process.

Backing Up the Anchor Server Database

- 1 Stop the anchorserver, either from the Administration Interface or using the following command::

```
$FASTSEARCH/bin/nctrl stop anchorserver
```

- 2 Back up the content of the directory *\$FASTSEARCH/data/anchorserver*.

- 3 Restart the anchorserver, either from the Administration Interface or using the following command::

```
$FASTSEARCH/bin/nctrl start anchorserver
```

Restoring the Anchor Server Database

- 1 Stop the anchorserver, either from the Administration Interface or using the following command::

```
$FASTSEARCH/bin/nctrl stop anchorserver
```

- 2 Restore the content of the directory *\$FASTSEARCH/data/anchorserver*.

- 3 Restart the anchorserver, either from the Administration Interface or using the following command::

```
$FASTSEARCH/bin/nctrl start anchorserver
```

Adding New Anchor Text Boosts to already Processed Documents

There is no way to apply new anchortext boost information to already indexed documents. If the anchortext boost information has changed enough to require it to be reconsidered, the only option is to re-feed documents through the document processing pipeline. Section *Re-feeding the Cached Content of the Crawler* describes how this can be done for already crawled documents.

Backing Up Index Data and Configuration

Introduction

Content fed into FAST InStream using the content APIs or a connector will first pass through document processing, which normally does not involve any storing of content to disk. Then documents will be fed to the index, which will store it to disk using the FAST internal indexer XML format called *fixml*. What parts of the original document to be maintained in the *fixml* format is largely dependent on the index profile and the document processing configuration.

The original document source cannot be reproduced from the *fixml* representation. In particular it is important to understand that only the data that needs to be available for presentation or result processing needs to be represented in its original form in the *fixml*, while content that is subject to indexing, but never will be used directly for display or result processing may not be retained by the search engine in the *fixml* source.

Normally, to be able to reprocess documents after changes in the document processing setup, one of the following assumptions must hold:

- Original content can be re-fed on demand from the content source. The File Traverser assumes this, and does not attempt to store the documents.
- The connectors maintain storage of the fetched documents. This is true for the crawler in the default configuration.

Backing Up Index Data

Backup of indexer nodes in a FAST InStream system is performed with the following procedure applied to each of the nodes to backup. To backup an entire indexing structure, this procedure must be applied to one of the nodes in each column in the system, each column consisting of nodes with identical data sets.

Main Procedure

The procedure is as follows:

- 1 Suspend indexing on the node to be backed up according to the procedure in section *Suspending Indexing on a Search Node* on page 8.
- 2 Backup the entire content of the `$FASTSEARCH/data/data_fixml` directory structure. This is mandatory.
- 3 Optional: backup the entire content of the `$FASTSEARCH/data/data_index` directory structure. See section *Optional Backup of Index Directory Data*.

- 4 Resume indexing according to procedure in Chapter 2 *Starting and Stopping FAST InStream*, section *Resuming Indexing on a Search Node*.

Optional Backup of Index Directory Data

The `$FASTSEARCH/data_index` directory contains the generated index used by the search engine. This index can be regenerated completely from the data in the `$FASTSEARCH/data_fixml` directory. Normally, backup of the index data is not necessary, but there could be cases where the system parameters are such that the time to recover is shorter when the index can be restored. Whether or not this is the case depends on the size of the resulting index, characteristics of the system such as available network bandwidth, backup medium and I/O performance.

It might well be the case that rebuilding the index from the fixml is faster than restoring it from backup. Another possible advantage of re-indexing (again dependent on the data, and the pattern of which it was fed into the system originally) is that an index constructed from scratch may be slightly smaller and better distributed, since an older index structure may have accumulated some disk space overhead due to fragmentation over time.

Backing Up Index Configuration

Indexer and search columns fetch most of their configuration from the Configuration Server, but need access to two files in the `$FASTSEARCH/etc` directory which contains “bootstrap” information such as where to find the Configuration Server. These files will be restored correctly by the installer if the index node is being reinstalled using the procedure in section *Node Installation Procedure* on page 68.

The files are:

```
searchrc-1.xml  
rtsplatformrc.xml
```

If more than one search column is running on a node, there will be files `searchrc-2.xml`, `searchrc-3.xml`... for each of the additional search columns.

Index Data Failover/Recovery

About this Chapter

This chapter covers failover and recovery procedures related to the indexed content. This includes the Search/Index nodes and the Status Server node. This chapter covers the following topics:

- Deployment Scenarios for High Availability
- Failover/recovery Without Index High Availability
- Failover/recovery With Index High Availability

Deployment Scenarios for High Availability

Introduction

High Availability and recovery needs for the indexed data and the search engines must be determined based on the application's needs. There are two basic aspects that must be analyzed:

- *Index Data High Availability.* In a single-node or single-row search configuration the search index data may be lost in case of hardware failure, and the content may need to be re-fed. By using two or more index rows the system may be restored if one node fails. The importance of this aspect depends on how easy it is and the time it will take to re-feed and re-index the content. The deployment scenarios 2 and 3 below provides such high availability configuration.

The recovery from an index node failure involves a manual procedure as described in this chapter.

- *Search High Availability.* In order to provide a high-availability system it is needed to have two search nodes and apply search interface high availability using a hardware load balancer in front of FAST InStream. Deployment scenario 3 provides such high availability configuration.

A more in-depth description of deployment scenarios can be found in the *Deployment Guide*.

Reference Scenarios

Figure 4-1 shows reference configurations that is used in the failover/recovery scenarios throughout this chapter.

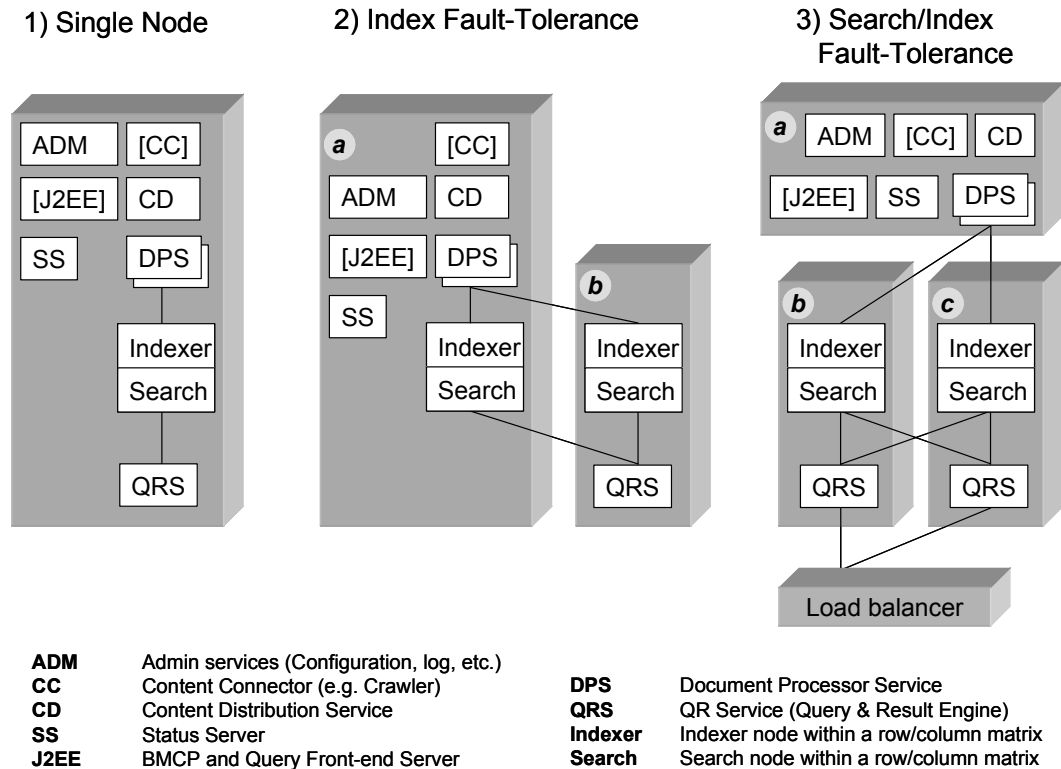


Figure 4-1 Reference Configurations

- 1** *Single Node.* In this scenario all the services is running on one server node.
- 2** *Index High Availability.* In this scenario two index/search rows are defined. Search high-availability search is not required, hence only one Query & Result Engine is required and no load-balancer.
- 3** *Search/Index High Availability.* In this scenario also search high-availability is required, and full high availability is added on the search interface side. The Load Balancer node refers to any 3rd party hardware fault-tolerance device (e.g. a Local Director) and is outside the scope of this document.

It is also possible to configure FAST InStream with search and indexer nodes on different servers. In this way you can have one indexer node that serves multiple search nodes. Such a scenario may be considered if it is important that all search rows are 100% in sync at any

time (using the same index). Configuration 2 and 3 above will contain the same content in the respective indices, but there may be temporary differences between the nodes due to incremental indexing, different load, etc. It is, however, recommended to consult FAST Professional Services before installing such a configuration.

Figure 4-2 shows an example System Management view from the Administration Interface for a multi-node installation equal to Scenario (3) in Figure 4-1.

FAST Data Search 4.0 - System Management - Microsoft Internet Explorer

Address: <http://oslres18.ad.fast.no:16000/admin/nodecontroller/index.php?shownode=oslres16.ad.fast.no>

fastdatasearch™ 4.0

Collection Overview | Document Processing | Search View | **System Management**

System Overview | Logs | Data Sources | Matching Engines

System Management Refresh | Help

System Monitor

3 Node Controllers are registered with the Config Server.

Host	As	CM	CS	CD	DP	EC	J2	LM	LS	NS	NC	QR	RI	RS	RT	SS	Ss	WS	DS
oslres15.ad.fast.no																			
oslres16.ad.fast.no																			
oslres18.ad.fast.no																			

[Show extended information for all nodes](#)

Node: oslres16.ad.fast.no

Control Panel

Node Commands:

Document Processor:

Crawler:

General system information

DataSearch location: e:\DATASE~1

Disk status: 2% used
216.3 GB free - 4.1 GB used - 220.4 GB total

Installed module list

Module name	Process name	PID	Status
Node Controller	nctrl	3416	Running
QRServer	qrserver	1132	Running
RTS Indexer	indexer	3240	Running
RTS Search	search-1	2844	Running
RTS Top Dispatcher	topfdispatch	3660	Running

Copyright © 2004 Fast Search & Transfer ASA

Figure 4-2 System Management View

Failover/recovery Without Index High Availability

This scenario describes a situation where a complete index and search column needs to be reconstructed after a fatal hardware crash. This applies to configuration (1) in Figure 4-1.

For mission critical deployments, where indexers are configured with high availability, this scenario would only be a last resort scenario in the case of multiple, simultaneous node failures, otherwise refer to section *Failover/recovery With Index High Availability* on page 37.

- 1 If the disk system is operative, you may back up the system that has failed to whatever extent is possible. Otherwise the latest backup of the system must be used.
- 2 Resolve the hardware problem that caused the failure.
- 3 Re-install the node from the install profile according to the procedure described in section *Node Installation Procedure* on page 68. This will re-install but not start the services on the recovered node.
- 4 Restore the latest Index data backup on the re-installed node. This implies restoring the following directory:

```
UNIX:      $FASTSEARCH/data/data_fixml
Windows:  %FASTSEARCH%\data\data_fixml
```

on the recovered node. This implies that documents received after the last backup will be lost, and a re-indexing of the data stored in fixml (pre-indexing format) files will be performed.

You may also decide to restore the actual index directory. Refer to section *Backing Up Index Data and Configuration* on page 29 for a more detailed discussion on the need for doing this. This involves restoring the following directory:

```
UNIX:      $FASTSEARCH/data/data_index
Windows:  %FASTSEARCH%\data\data_index
```

- 5 Restore other data from the latest backup. Refer to the procedures in Chapter 3 *Backing Up and Restoring FAST InStream*.
- 6 Start the necessary FAST InStream services on the recovered node. This is performed according to the procedure in section *Starting FAST InStream* on page 4.
Refresh the administrator interface **System Overview** page. The new node should now have appeared.
- 7 Trigger recovery of the Status Server as described in section *Backing Up and Restoring the Status Server* on page 22. This must be performed in order to ensure consistency between the index data and Status Server data related to documents.
- 8 Resume all connector feeds.

- 9 As documents may have been lost from the time of the last backup, it may be required to perform a full re-feed or re-crawl from the content source. Refer to section *Backing Up and Restoring Crawler Configuration and Data* on page 18.
- 10 If backup has been restored and changes have been made to rank tuning after the time of the backup, the potential effect of those changes must be deployed to the backed-up index. Trigger reprocessing of BMCP rank tuning information according to procedure in section *Backing Up and Restoring the BMCP Database (Rank Tuning)* on page 24.

Failover/recovery With Index High Availability

Introduction

The described scenarios shows how to recover from a search/indexer node failure in a system according to configuration 2 and 3 in Figure 4-1. One of the nodes in a column needs to be reconstructed after a fatal hardware crash, and there exist one or more backup nodes within the column (multiple rows).

The procedure includes suspending the content feeding and indexing, backing up the Index data from the working row node and restoring the failed node with the index data from the working node. In this way it is ensured that the recovered node becomes in sync with the working node after recovery.

Prerequisites

In order to handle a single node error it is not required to have a full backup of the node. The only prerequisite is that you have backed up the Installation Profile. The installation profile is automatically generated upon normal installation. Refer to section *Node Installation Procedure* on page 68 for details.

Note! You should also perform a regular backup of one row node within each column regularly, in case of a more fatal hardware error involving more than one node.

Single Index/Search Node Failure

Dedicated Search/Index Node

This scenario describes a search/indexer node failure in a system according to scenario 2 (failure on node *b*) and scenario 3 (failure on node *a* or *b*) in Figure 4-1. One of the nodes in a column needs to be reconstructed after a fatal hardware crash, and there exist one or more backup nodes within the column (multiple rows).

- 1 Resolve the hardware problem that caused the failure.
- 2 Re-install the node from the install profile according to the procedure described in section *Node Installation Procedure* on page 68. This will re-install but not start the services on the recovered node.

3 Suspend feeding and indexing.

- a** Stop feeding from the content source as described in section *Suspending the Connector Feed* on page 6.
 - b** Give the document processing some time to finish outstanding batches submitted by the connector(s). When using the Enterprise Crawler you can monitor this using the procedure as described in section *Suspending the Connector Feed* on page 6.
 - c** Suspend indexing on all working indexing nodes within the column as described in section *Suspending Indexing on a Search Node* on page 8.
- 4** Create a backup of the other index node in the installation according to the procedure in section *Backing Up Index Data and Configuration* on page 29.
- 5** Restore the latest Index data backup on the re-installed node. This implies restoring the following directory:

```
UNIX:      $FASTSEARCH/data/data_fixml
Windows:   %FASTSEARCH%\data\data_fixml
```

on the recovered node. This implies that documents received after the last backup will be lost, and a re-indexing of the data stored in fixml (pre-indexing format) files will be performed.

You may also decide to restore the actual index directory. Refer to section *Backing Up Index Data and Configuration* on page 29 for a more detailed discussion on the need for doing this. This involves restoring the following directory:

```
UNIX:      $FASTSEARCH/data/data_index
Windows:   %FASTSEARCH%\data\data_index
```

- 6** Start the necessary FAST InStream services on the recovered node. This is performed according to the procedure in section *Starting FAST InStream* on page 4.
- Refresh the administrator interface **System Overview** page. The new node should now have appeared.
- 7** Resume indexing according to section *Resuming Indexing on a Search Node* on page 10.
- 8** Resume all connector feeds.

Search/Index Node also including a Content Distributor

This scenario describes a search/indexer node failure in a system according to scenario 2 (failure on node *a*) in Figure 4-1. The failed node also includes the Content Distributor and the Status Server. One of the nodes in a column needs to be reconstructed after a fatal hardware crash, and there exist one or more backup nodes within the column (multiple rows).

- 1 Resolve the hardware problem that caused the failure.
- 2 Re-install the node from the install profile according to the procedure described in section *Node Installation Procedure* on page 68. This will re-install but not start the services on the recovered node.
- 3 Create a backup of the other index node in the installation according to the procedure in section *Backing Up Index Data and Configuration* on page 29.
- 4 Restore the latest Index data backup on the re-installed node. This implies restoring the following directory:

```
UNIX:      $FASTSEARCH/data/data_fixml
Windows:  %FASTSEARCH%\data\data_fixml
```

on the recovered node. This implies that documents received after the last backup will be lost, and a re-indexing of the data stored in fixml (pre-indexing format) files will be performed.

You may also decide to restore the actual index directory. Refer to section *Backing Up Index Data and Configuration* on page 29 for a more detailed discussion on the need for doing this. This involves restoring the following directory:

```
UNIX:      $FASTSEARCH/data/data_index
Windows:  %FASTSEARCH%\data\data_index
```

- 5 Restore other data from the latest backup. Refer to the procedures in Chapter 3 *Backing Up and Restoring FAST InStream*.
- 6 Start the necessary FAST InStream services on the recovered node. This is performed according to the procedure in section *Starting FAST InStream* on page 4

Make sure that no content feed is started. In case of the crawler you may need to stop it after running 'nctrl start'.

Refresh the administrator interface **System Overview** page. The new node should now have appeared.

- 7 Resume indexing according to section *Resuming Indexing on a Search Node* on page 10.
- 8 Recover the Status Server as described in section *Backing Up and Restoring the Status Server* on page 22.
- 9 Resume all connector feeds.

Chapter 5

Applying Run-Time Deployment Changes

About this Chapter

FAST InStream is designed for highly flexible scalability. Typically, as the use of the system grows, at some point you will need to add more hardware capacity to allow further distribution of tasks within the system.

This chapter provides an overview of alternatives and main procedures for applying deployment changes in a running FAST InStream system.

Note! This chapter is complementary to the *Deployment Guide*, which provides overall consideration information on when to scale a system. In particular, you may refer to Chapter 4 *Performance Optimization* in the *Deployment Guide*. While the *Deployment Guide* provides a means to locate bottlenecks in a running system, this chapter provides you with specific procedures on how to change the deployment at run-time, typically without interrupting the search service.

This chapter includes:

- Capacity Scaling Dimensions
- Increasing Query Capacity and High Availability
- Increasing the Content Volume Capacity
- Increasing Document Feed/Processing Capacity
- Configuring new Search Nodes in the FAST Configuration Server
- Adding a Top-Level Dispatcher to a Single Search Node System

Capacity Scaling Dimensions

Typically, a FAST InStream installation will scale and saturate along one or more of the following three dimensions:

- *Content volume:*

Content volume measures the amount of content to be handled by the system. Content characteristics play also an important role.

For a detailed procedure, refer to section *Increasing the Content Volume Capacity* on page 50.

- *Query rate:*

Query rate measures the number of queries per second that the system must be able to handle.

If the query rate is exceeding the capacity of the search service, you need to increase the query rate capacity of the search service. For a detailed procedure, refer to section *Increasing Query Capacity and High Availability* on page 43.

- *Content dynamics:*

Content dynamics measure how the content changes and how these changes must be reflected in the system.

If the document feed load is moving towards the processing capacity of the system, you need to identify the bottleneck, which can be in several places. Refer to section *Increasing Document Feed/Processing Capacity* on page 54 for further discussion on this topic.

As discussed in the *Deployment Guide*, the actual configuration planning is a complicated task with many individual parameters to consider. In most cases it is recommended to consult FAST Professional Services during the initial planning, and you also need to ensure that you address the right component to scale. E.g. do not add a new search row if the problem eventually might be on the Query API client side.

Increasing Query Capacity and High Availability

Loading the system beyond query rate capacity will cause a backlog of queries to build up. This yields higher query latencies and possibly disruption of service. Depending on where in the system the bottleneck is, the backlog may in turn cause timeouts and retransmits, which in turn generates an even higher load on the search service, with resulting degradation of service. With this in mind, it is recommended that you plan for increasing query loads and upgrade the system accordingly.

The query rate capacity is mainly limited by:

- The number of search rows. Multiple search rows provides a linear scaling of QPS within the Search Engines.
- The number of Query and Result Engines (QRServers). This is especially important when using result-processing features.
- Query API client performance. For certain features the Query API performs a substantial amount of parsing and processing. This includes *Result Clustering* and *Navigators* with many buckets. This is mainly related to API client design and not further described in this document.
- Feature set. The actual feature set has strong impact on the effective QPS. Result-side features such as *Dynamic Duplicate Removal*, *Result Clustering* and *Result-side Navigators* will add substantially to the load on the QRServers. Other features such as *Deep Navigators* and *Full Wildcard Support* adds more load on the search nodes.

Consult the *Deployment Guide* for further details on how to analyze the system to discover current bottlenecks. Based on this analysis the following deployment changes may be considered:

- *Adding a new Search Row*
- *Adding a new Query and Result Engine*

These deployment changes are further described throughout this section.

Adding a new Search Row

If the number of documents per node is still well below capacity, but query volume is approaching a level where the existing rows are getting saturated, you need to add a new search row. Depending on your initial configuration, this may be achieved in two ways:

- *Adding a Combined Index and Search Row*. This is normally the recommended solution, where indexer and search is co-located on the same server node. This also provides indexing high availability as each row has a dedicated indexer.

- *Adding a Dedicated Search Row.* This is based on an initial configuration with indexer and search is located on different server nodes. This approach may be used for performance optimization in larger installations. FAST recommends that you consult FAST Professional Services before deploying such a configuration initially.

Adding a Combined Index and Search Row

To ensure a balanced system, new rows need to be similar to the existing rows. This means that if the existing rows are combined search and indexer nodes, the new row also needs to be combined.

Figure 5-1 shows the configuration change as described in this section:

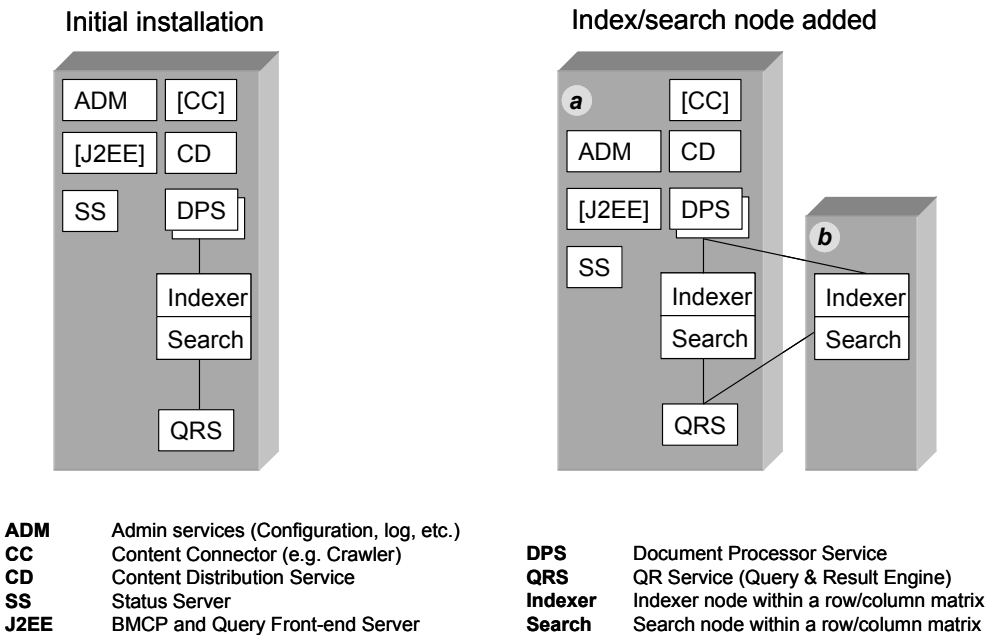


Figure 5-1 Adding a Combined Index and Search Row

Note! If your existing system has dedicated search respectively indexer rows, refer to section *Increasing the Content Volume Capacity* on page 50 for information on how to extend query capacity.

To add a combined index and search row, complete the following procedure:

- 1 If the current system is configured with only one search node, you need to configure a top-level dispatcher.

Refer to section *Adding a Top-Level Dispatcher to a Single Search Node System* on page 59 for a detailed procedure.

- 2 Install the new node(s) needed for the new row.
 - a If you have installed your initial FAST InStream system using the OEM Installer (refer to the FAST InStream SDK) it is normally preferred to use the procedure as described in Chapter 7 *Adding a new Node using the OEM Installer*. Refer to section *Convert a single-node installation into a node in an existing multi-node installation* on page 74 for a detailed procedure.
 - b If you have installed your FAST InStream system using the standard FAST InStream Installer you should use the procedures as described in Chapter 6 *Installing Nodes from an Install Profile*. Refer to section *Configuring new Nodes in the Install Profile* on page 71 for details on configuring the Install Profile for this configuration.
- 3 Configure the new row on the Admin Node. Refer to section *Configuring a New Search Row in the FAST Configuration Server* on page 57.
- 4 Ensure that the license file is configured properly on the node. Refer to section *Multi-node License File Considerations* on page 62.
- 5 Stop content feeding on the existing FAST InStream system.

Refer to section *Suspending the Connector Feed* on page 6 for a detailed procedure.
- 6 Stop the Status Server

```
$FASTSEARCH/bin/ncctl stop statusserver (Unix)
%FASTSEARCH%\bin\ncctl stop statusserver (Windows)
```
- 7 Stop indexing on all nodes by running the following command:

```
$FASTSEARCH/bin/ncctl stop indexer (Unix)
%FASTSEARCH%\bin\ncctl stop indexer (Windows)
```

on each node.

This will ensure that indexing is stopped in a controlled way, while searches still will be executed.
- 8 Stop search on all indexer and search nodes by running the following command:

```
$FASTSEARCH/bin/ncctl stop search-1 (Unix)
%FASTSEARCH%\bin\ncctl stop search-1 (Windows)
```

on each node.
- 9 Copy the entire content of the *\$FASTSEARCH/data/data_fixml* (Unix) or *%FASTSEARCH%\data\data_fixml* (Windows) directory from each of the old indexer nodes to the corresponding new node, that means the node with the same position in the row.
- 10 Start up the new node according to the procedure described in *Starting FAST InStream*

on page 4.

When the search processes start, they will contact their respective indexers and receive the latest index. When that index is ready, search will automatically be redistributed across the entire columns including the nodes in the new row.

11 Start the Status Server.

At startup, the Status Server it will resolve available indexers.

Refer to section *Backing Up and Restoring the Status Server* on page 22 for a detailed procedure.

12 Restart content feeding by restarting all processor servers.

13 Restart any Connector that might be deployed in your system. Content feeding will commence as soon as the Status Server is done with recovery.

Adding a Dedicated Search Row

To ensure a balanced system, new rows need to be similar to the existing rows. This means that if the existing system has dedicated search respectively indexer rows, new rows also need to be dedicated.

Figure 5-2 shows the configuration change as described in this section:

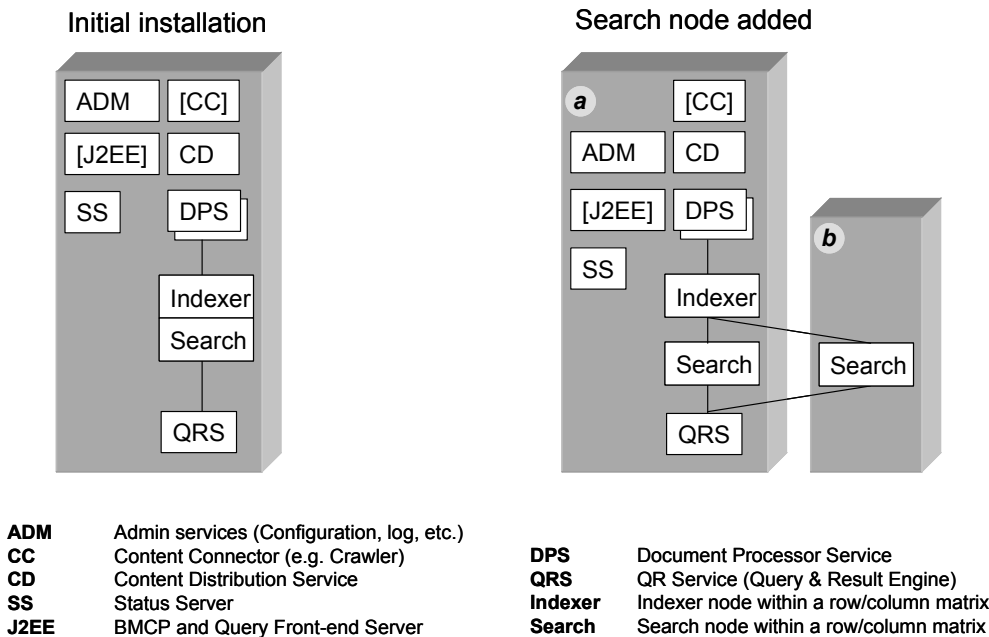


Figure 5-2 Adding a Dedicated Search Row

Note! If the existing system consists of combined search respectively indexer rows, or if you want to extend index capacity, refer to section *Adding a Combined Index and Search Row* on page 44. For information on how to add new search rows to combined index and search nodes, refer to section *Adding a Dedicated Search Row* on page 46.

To add a new search row, complete the following procedure:

- 1 If the current system is configured with only one search node, you need to configure a top-level dispatcher.

Refer to section *Adding a Top-Level Dispatcher to a Single Search Node System* on page 59 for a detailed procedure.

- 2 Install the new node(s) needed for the new row.
 - a If you have installed your initial FAST InStream system using the OEM Installer (refer to the *FAST InStream SDK*) it is normally preferred to use the procedure as described in Chapter 7 *Adding a new Node using the OEM Installer*. Refer to section *Convert a single-node installation into a node in an existing multi-node installation* on page 74 for a detailed procedure.
 - b If you have installed your FAST InStream system using the standard FAST InStream Installer you should use the procedures as described in Chapter 6 *Installing Nodes from an Install Profile*. Refer to section *Configuring new Nodes in the Install Profile* on page 71 for details on configuring the Install Profile for this configuration.
- 3 Configure the new row on the Admin Node. Refer to section *Configuring a New Search Row in the FAST Configuration Server* on page 57.
- 4 Ensure that the license file is configured properly on the node. Refer to section *Multi-node License File Considerations* on page 62.
- 5 Start up the new node according to the procedure described in *Starting FAST InStream* on page 4.

When the search processes start, they will contact their respective indexers and receive the latest index. When that index is ready, search will automatically be redistributed across the entire columns including the nodes in the new row.

Adding a new Query and Result Engine

The need for adding more Query and Result Engines (qrserver processes) in the system arises when the system has a query load that saturates the current qrserver(s). Adding a second Query and Result Engine may also be done in order to provide high availability search. An example of such a deployment can be found in scenario (3) in the figure in section *Reference Scenarios* on page 33.

Note! Another common reason for a limited query rate may be that there are not enough search rows. For information on how to extend the number of search rows, refer to section *Increasing Query Capacity and High Availability* on page 43.

The following procedure describes how to add a new Query and Result Engine on an existing node within your installation, e.g. a newly installed second search row node. To add a new query and result processor, complete the following procedure:

- 1 Select the node on which to provide the new query and result processor.
- 2 Copy the content of the `$FASTSEARCH/etc/qrserver` (Unix) or `%FASTSEARCH%\etc\qrserver` (Windows) directory from one of the existing query and result processor nodes to the same location on the new node.
- 3 Enable the new query and result server configuration entry in the file `NodeConf.xml` located in the `$FASTSEARCH/etc/` (Unix) or `%FASTSEARCH%\etc\` (Windows) directory by adding it to the start list:

In the section:

```
<startorder>
...
</startorder>
```

add the following line:

```
<proc>qrserver</proc>.
```

- 4 In general multiple query and result processors can run towards the same dispatcher. However, it is recommended that you add an extra top-level dispatcher for the new query and result processor.

Refer to section *Adding a Top-Level Dispatcher to a Single Search Node System* on page 59 for a detailed procedure.

- 5 If you choose not to add a separate top-level dispatcher, you need to start the new query and result processor explicitly by running the following command:

```
nctrl start qrserver
```

- 6 Finally, you need to spread the query load across the extended set of query and result

processors. To do this, you need additional components such as a local director or a custom load balancer in the front end. For information on how to implement these components, contact FAST Professional Services.

Increasing the Content Volume Capacity

If the number of documents served is increasing beyond recommended sizing you need to add new search columns. Refer to Chapter 4 *Performance Optimization*, section *Content Volume Performance* in the *Deployment Guide* for detailed sizing recommendations.

The content volume scaling implies configuring new indexer nodes and re-partitioning existing data across these nodes. Depending on your initial configuration, this may be achieved in two ways:

- *Adding a Combined Index and Search Column.* This is normally the recommended solution, where indexer and search is co-located on the same server node.
- *Adding a new column using dedicated Index Nodes and Search Nodes.* This is based on an initial configuration where indexer and search is located on different server nodes, and may be used for performance optimization in larger installations. Consult FAST Professional Services before deploying such a configuration.

Adding a Combined Index and Search Column

To ensure a balanced system, new columns should be similar to the existing columns. This means that if the existing columns are combined search and indexer nodes, the new column also needs to be combined.

The configuration change as described in this section is equal to Figure 5-1 in section *Adding a Combined Index and Search Row* on page 44, even if the actual role split between the two nodes are different.

To add a new search column, complete the following procedure:

- 1 If the current system is configured as a single search node system, you need to configure an additional top-level dispatcher.
Refer to section *Adding a Top-Level Dispatcher to a Single Search Node System* on page 59 for a detailed procedure.
- 2 Install the new node(s) needed for the new column.
 - a If you have installed your initial FAST InStream system using the OEM Installer (refer to the FAST InStream SDK) it is normally preferred to use the procedure as described in Chapter 7 *Adding a new Node using the OEM Installer*. Refer to section *Convert a single-node installation into a node in an existing multi-node installation* on page 74 for a detailed procedure.
 - b If you have installed your FAST InStream system using the standard FAST InStream Installer you should use the procedures as described in Chapter 6 *Installing Nodes from an Install Profile*. Refer to section *Configuring new Nodes*

in the *Install Profile* on page 71 for details on configuring the Install Profile for this configuration.

- 3 Ensure that the license file is configured properly on the node. Refer to section *Multi-node License File Considerations* on page 62.

- 4 Stop content feeding.

Refer to section *Suspending the Connector Feed* on page 6 for a detailed procedure.

- 5 Stop the Status Server

```
$FASTSEARCH/bin/nctrl stop statusserver (Unix)
%FASTSEARCH%\bin\nctrl stop statusserver (Windows)
```

- 6 Stop indexing on all nodes by running the following command:

```
$FASTSEARCH/bin/nctrl stop indexer (Unix)
%FASTSEARCH%\bin\nctrl stop indexer (Windows)
```

on each node.

This will ensure that indexing is stopped in a controlled way, while searches still will be executed.

- 7 To ensure even load distribution, you need to manually redistribute a suitable amount of files from the *\$FASTSEARCH/data/data_fixml* (Unix) or *%FASTSEARCH%\data\data_fixml* (Windows) directory on each of the existing nodes over to the new column.

When moving to X columns, you should move 100/X % of the documents from the existing columns to the new column. You can determine the number of documents in a fixml directory using the command:

```
$FASTSEARCH/fidxobj countdocs <folder name>
```

To distribute the documents, complete the following procedure:

- a On the new node, create a directory called *data/data_fixml* (Unix) or *data\data_fixml* (Windows).
- b Create the necessary Set_0000X catalogs within this directory. The created catalogs must comply with the configured maximum number of files per set. The maximum number of files per set is 250, which means that no Set_0000X catalog must contain more than 250 fixml files.
- c Copy selected fixml files and their associated *.magic* file from the old nodes to the new node. Make sure the copied files have unique file names. Rename or renumber the files if necessary.

The numbering does not need to be sequential, that means there may be gaps. Different sets or nodes may have conflicting names.

- d Delete the copied files from the old nodes.

- 8 On the old nodes, delete all non-fixml files or directories in the *\$FASTSEARCH/data/data_fixml* (Unix) or *\$FASTSEARCH\data\data_fixml* (Windows) directory on each of the existing nodes. In general, these files are part of a db directory. You need to delete this directory with all its content and possibly any *.dat* or *.tmp* files, leaving only the *Set_nnnnnn* directories containing the fixml files, quite similar to what the content will be on the newly constructed nodes constituting the new column.

- 9 Configure an extra column in the configuration server.

Refer to section *Configuring a New Search Column in the FAST Configuration Server* on page 56 for a detailed procedure.

- 10 Restart indexing on all indexer nodes, including the new column.

Refer to section *Resuming Indexing on a Search Node* on page 10 for a detailed procedure.

- 11 On each indexer node, trigger a resetindex by running the **rtsadmin** command as shown in the following example:

The rtsadmin program accepts the following parameters:

- **Name service node**
- **Name service port**
- **Search cluster name**
- **Column number**
- **Row number**

The following example is based on a two row - two column search cluster named 'webcluster' with nameservice on node named 'ns'.

```
$FASTSEARCH/bin/rtsadmin ns 31099 webcluster 0 0 resetindex
$FASTSEARCH/bin/rtsadmin ns 31099 webcluster 1 0 resetindex
$FASTSEARCH/bin/rtsadmin ns 31099 webcluster 0 1 resetindex
$FASTSEARCH/bin/rtsadmin ns 31099 webcluster 1 1 resetindex
```

- 12 Start the Status Server.

At startup, the Status Server will discover the new column.

Refer to section *Backing Up and Restoring the Status Server* on page 22 for a detailed procedure.

- 13 Trigger the Status Server recovery.

This will force the Status Server to adjust its routing tables to the new distribution of the documents in the index nodes.

Refer to section *Backing Up and Restoring the Status Server* on page 22 for a detailed procedure.

- 14 Restart content feeding by restarting all processor servers.

- 15 Restart any Connector that might be deployed in your system. Content feeding will commence as soon as the Status Server is done with recovery.

Note! This procedure allows the system to serve searches uninterruptedly while the deployment changes are applied. However, consistency of data is not ensured for a short period of time, when the new column gets available and some of the old search columns still serve searches from their old indexes. This may lead to some searches resulting in duplicates. Once all old nodes are done with the `resetindex` operation and have switched to the new index, data consistency is restored.

If data consistency at any given time is critical for your implementation, you might need to stop all search nodes while the nodes are reindexing and restart the search processes once all search nodes have switched to the new index. Refer to section *Running a Sequential Shutdown* on page 5 for details on how to stop search

Increasing Document Feed/Processing Capacity

In order to increase content feed and document processing capacity you may need to perform one of the following tasks:

- For large-scale crawler installations it may be recommended to configure a multi-node crawler configuration. Refer to the *Configuration Guide* for information on how to add crawler nodes.
- The Document Processing nodes (ProcessorServer) running the document processing pipelines is a single-threaded solution. Hence, it is recommended to run multiple processor server instances on each server node.

Refer to the Configuration Guide for information on how to add document processing nodes.

- In large-scale configurations it may be necessary to tune the Status Server resource allocation. Refer to section *Performance Tuning of the FAST InStream Status Server* for further details.

Consult the *Deployment Guide* for further details on scaling considerations.

Performance Tuning of the FAST InStream Status Server

The Status Server keeps track of certain information about every document being processed by the system. For small installations, it is most efficient for the Status Server to keep this information in memory. For larger installations with many documents, the Status Server supports storing this information on disk.

Use the `$FASTSEARCH/bin/ssconfiglient` utility to configure the Status Server. Running this program without parameters provides information on usage. Configuration is specified for individual collections. The most important parameters are:

Table 5-1 ssconfiglient parameters

Parameter	Description
<code>--mem_min</code>	The initial size of the memory structure for maintaining document information. The size is measured in number of documents.
<code>--mem_max</code>	The maximum size of the memory structure for maintaining xxx document information. When more than this number of documents is being processed, the Status Server will switch to disk-based data structures.
<code>--mem_ex</code>	The factor used when increasing the size of the memory structure when it becomes too small to maintain information about all documents in the system. The size will not expand beyond <code>--mem_max</code> .

Table 5-1 ssconfigclient parameters

Parameter	Description
--disk_min	The initial size of the disk based storage structure.
--disk_ex	The factor used when increasing the size of the disk based structure when it becomes too small to maintain information about all documents in the system.

The memory structure will always be persisted to disk when the Status Server is shut down in a controlled manner. If shutdown is uncontrolled (for instance when the machine is powered off), it will perform a recovery operation to reconstruct the information the next time it is started. Refer to section *Backing Up and Restoring the Status Server* on page 22.

About 24 bytes are needed per document being processed by the system. The data structures maintained by the Status Server cannot utilize more than about 70% of allocated space and still be efficient. This means that a structure that has allocated space for 1.000.000 documents can efficiently hold information about 700.000 documents and consume about 24MB of memory.

Configuring new Search Nodes in the FAST Configuration Server

Configuring a New Search Column in the FAST Configuration Server

To configure a new search column in the Configuration Server, complete the following procedure:

- 1 Stop the Configuration Server by running the following command from the command line:

```
$FASTSEARCH/bin/nctrl stop configserver (Unix)
$FASTSEARCH\bin\nctrl stop configserver (Windows)
```

- 2 Edit the *CSCfg.xml* file located in the *\$FASTSEARCH/etc/* (Unix) or *%FASTSEARCH%\etc* (Windows) directory.

Add a new column entry in the `<search_clusters>` section by copying the existing `<column>` entry.

Use the following attribute values for the new `<column>` entry:

Table 5-2 `<column>`- Attribute Specifications

Attribute	Description
host	Fully qualified host name for the new node.
port	Port number used for this search node. Set to same value as for existing <code><column></code> entry.
mode	Set to same value as for existing <code><column></code> entry.
partition_id	Set <code>partition_id</code> to one more than the largest <code>partition_id</code> already in the file.
ft_mode	Set to same value as for existing <code><column></code> entry.
docapiport	Port number to the internal search node document API. Set to same value as for existing <code><column></code> entry.
rowid	The second row (first <code><backup></code> entry) shall have <code>rowid="1"</code> . If 2 or more rows are already defined, set the <code>rowid</code> to the next available value ("2" if this is the third row defined for the column).

The following example shows a cluster entry where a second column is added. The added column entry is highlighted:

```
<search_clusters>
  <cluster name="webcluster">
    <column host="node01.yourdomain.com" port="15674" mode="NORMAL"
      partition_id="0" ft_mode="0" docapiport="15500" />
    <column host="node01.yourdomain.com" port="15674" mode="NORMAL"
      partition_id="1" ft_mode="0" docapiport="15500" />
    </column>
    <dispatcher host="node01.yourdomain.com" port="15100"/>
  </cluster>
</search_clusters>
```

If the existing column includes multiple rows you should add the same number of rows in the new column. Refer to the next section *Configuring a New Search Row in the FAST Configuration Server*

- 3 Start the Configuration Server by running the following command from the command line:

```
$FASTSEARCH/bin/ncctl start configserver (Unix)
%FASTSEARCH%\bin\ncctl start configserver (Windows)
```

Configuring a New Search Row in the FAST Configuration Server

To configure a new search row in the Configuration Server, complete the following procedure:

- 1 Stop the Configuration Server by running the following command from the command line:

```
$FASTSEARCH/bin/ncctl stop configserver (Unix)
$FASTSEARCH\bin\ncctl stop configserver (Windows)
```

- 2 Edit the *CConfig.xml* file located in the *\$FASTSEARCH/etc/* (Unix) or *%FASTSEARCH%\etc* (Windows) directory.

Add a new row entry in the `<search_clusters>` section by adding a `<backup>` entry after the `<column>` entry. If two or more rows are already defined you can copy the existing `<backup>` entry and only modify the rowid attribute.

Use the following attribute values for the new <backup> entry:

Table 5-3 <backup>- Attribute Specifications

Attribute	Description
host	Fully qualified host name for the new node.
port	Port number used for this search node. Normally this should be set to the same value as the corresponding attribute in the <column> entry (or existing <backup> entry).
docapiport	Port number to the internal search node document API. Normally this should be set to the same value as the corresponding attribute in the <column> entry (or existing <backup> entry).
rowid	The second row (first <backup> entry) shall have rowid="1". If 2 or more rows are already defined, set the rowid to the next available value ("2" if this is the third row defined for the column).

The following example shows a cluster entry where a second row is added. The added/changed data is highlighted:

```
<search_clusters>
  <cluster name="webcluster">
    <column host="node01.yourdomain.com" port="15674" mode="NORMAL"
      partition_id="0" ft_mode="0" docapiport="15500">
      <backup host="node02.yourdomain.com" port="15674"
        docapiport="15500" rowid="1"/>
    </column>
    <dispatcher host="node01.yourdomain.com" port="15100"/>
  </cluster>
</search_clusters>
```

- 3 Start the Configuration Server by running the following command from the command line:

```
$FASTSEARCH/bin/nctrl start configserver (Unix)
%FASTSEARCH%\bin\nctrl start configserver (Windows)
```

Adding a Top-Level Dispatcher to a Single Search Node System

Top-level dispatchers must be present in any system that has or is to have more than one search node. The installer will automatically configure a top-level dispatcher when installing a multi search node system. If you want to upgrade a single search node system to a multi search node system, you must add a top-level dispatcher.

Note! A top-level dispatcher works fine also with only a single node, but adds a small amount of overhead to the system. Consequently, you may add a top-level dispatcher even if you only have a single node system to prepare for future extensions of the system.

The top-level dispatcher typically runs on the same node as the Query and Result Engine.

Note! If you have multiple Query and Result Engine nodes, you need to add a top-level dispatcher to each of the nodes.

To add a top-level dispatcher to a node, complete the following procedure:

- 1 In the *\$FASTSEARCH/etc/* (Unix) or *%FASTSEARCH%\etc* (Windows) directory, create a file called *searchrc-dispatch.xml* containing the following lines:

```
<search>
  <common
    hostname           = "node01.yourdomain.com"
    baseport           = "<baseport number>"
    namingservice      = "adminnode.domain.tld:16099"
    columnrowid        = "0-0"
    clustername        = "webcluster"
    mode               = "dispatch"
    dispatchonlyrow    = "false"
    debuglog           = "false"
  />
</search>
```

using the following values for the individual attributes:

Table 5-4 searchrc-dispatch.xml Attributes and Values

Attribute	Description of the value to be used
hostname	The qualified hostname that is used by the Query and Result Engine (QRServer).
baseport	The number of the baseport for this component (not the FAST InStream base port!). You determine this by using the following formula: <baseport number of the initial installation> + 2150. The default baseport number of the FAST InStream installation is 13000.
namingservice	The qualified host name of the CORBA name service. The port number to be used is determined using the following formula: <baseport number of the initial installation> + 3099. The default baseport number of the FAST InStream installation is 13000.
columnrowid	Always set to "0-0" for a top-level dispatcher
clustername	The name of the cluster the top-level dispatcher is to be assigned to. The default clustername for any single cluster installation is "webcluster".
mode	The role of the component. (search node or dispatcher). Set to "dispatch"
dispatchonlyrow	Always set to "false"
debuglog	Only for internal debugging. Set to "false"

Note! If the *searchrc-dispatch.xml* file already exists, you only need to make sure it refers to the correct cluster name.

- 2 Enable the top-level dispatch configuration entry in the file *NodeConf.xml* located in the *\$FASTSEARCH/etc/* (Unix) or *%FASTSEARCH%\etc* (Windows) directory of the node the top-level dispatcher is to run on. You do this by adding the entry to the start list:

In the section:

```
<startorder>
...
</startorder>
```

add the following line:

```
<proc>topfdispatch</proc>
```


right before the entry `<proc>search-1</proc>`.

- 3 Make the node controller aware of the new configuration. To do this, perform the following command:

```
nctrl reloadcfg
```

- 4 Start the new top level dispatcher by running the following command from the command line:

```
nctrl start topfdispatch
```

- 5 Edit the *webcluster.spec* file located in the *\$FASTSEARCH/etc/qrserver/* (Unix) or *%FASTSEARCH%\etc\qrserver* (Windows) directory to reflect the new top-level dispatch host:

Replace the existing port number, which is the port number of the single search node, by the port number of the new top-level dispatcher. This port number is calculated based on the following formula:

`<baseport number of the initial installation> + 2150`

with `<baseport number>` representing the baseport number you indicated during the initial system installation. By default, this number is 13000.

Note! If you specified a different cluster name during installation, the name of the file to be changed uses this cluster name.

- 6 Ensure that the *CSConfig.xml* file contains a corresponding `<dispatcher>` entry as indicated in section *Configuring new Search Nodes in the FAST Configuration Server* on page 56.
- 7 Restart the query and result processor by running the following command:

```
nctrl restart qrserver
```

Now any query is directed to the top-level dispatcher, which will automatically adapt to search node layout changes.

Multi-node License File Considerations

In a distributed configuration the license manager uses the master license file located on the host where the license manager is running. It is this master license file that includes the detailed feature and capacity license entries.

The license manager host is normally the same as the host where the FAST Configuration Server is running. Each of the other nodes in the system must also have a license file, but this is only used in order to tell the processes where to find the license manager.

On all other nodes than the administration node the license file only needs the two first lines in the license file,

```
SERVER <fully qualified host name> ANY  
USE_SERVER
```

The license file (fastsearch.lic) shall be located in the \$FASTSEARCH/etc directory on each node.

If another non-admin node is available, simply copy the license file from this node. Otherwise the license file from the admin node can also be used, although the actual FEATURE lines in the license file is not relevant in this case (always verified by the license manager on the admin node).

Also ensure that the environment variables are set according to the procedure in Chapter 1 *Environment Variables*. The `LM_LICENSE_FILE` environment variable must point to the installed license file, `$FASTSEARCH/etc/fastsearch.lic`.

Chapter 6

Installing Nodes from an Install Profile

About this Chapter

This chapter provides procedures for installing new nodes in a FAST InStream system using an Install Profile.

In general the installation steps described in this chapter configures individual nodes in the system with given services. However, the overall procedures for adding new nodes in a FAST InStream system involves additional steps as described in Chapter 5 *Applying Run-Time Deployment Changes*.

Note! This chapter is complementary and provides sub-procedures to Chapter 5 *Applying Run-Time Deployment Changes*.

This chapter includes:

- Install Profile Overview
- Node Installation Procedure
- Configuring new Nodes in the Install Profile

Install Profile Overview

The Install Profile defines the configuration setup that is created when installing FAST InStream. This configuration file allocates hosts to services and can be used in order to re-install a recovered node or install a new node from scratch. The file is named *InstallProfile.xml* and is by default saved in the FAST InStream installation directory.

Table 6-1 provides an example Install Profile for a multi-node installation with 2 search rows:

Table 6-1 Install Profile Example

Install Profile Elements/Attributes	Description
<pre><?xml version="1.0"?> <system-configuration name="FAST InStream Installer generated configuration" version="4.1"></pre>	
<pre><host-set> <host id="node1.example.com"> <property name="hostname" value="node1.example.com"/> <property name="platform" value="Linux"/> <property name="username" value="username"/> <property name="install-dir" value="/home/username/fast"/> <property name="temp-dir" value="/tmp"/> <property name="remote-cmd" value="SSH2"/> <property name="baseport" value="13000"/> </host></pre>	<p><host> entry for each host. id typically set to the hostname</p> <p>hostname: Fully qualified host name platform: OS platform identification. username: Install user name install-dir: Installation directory</p> <p>temp-dir: For temporary files remote-cmd: For executing commands on the remote host baseport: Base port for the installation, default 13000.</p>
<pre><host id="node2.example.com"> <property name="hostname" value="node2.example.com"/> <property name="platform" value="Linux"/> <property name="username" value="username"/> <property name="install-dir" value="/home/username/fast"/> <property name="temp-dir" value="/tmp"/> <property name="remote-cmd" value="SSH2"/> <property name="baseport" value="13000"/> </host></pre>	<p>Second host defined.</p>

Table 6-1 Install Profile Example

Install Profile Elements/Attributes	Description
<pre> <administration-set> <configuration-service id="CS1" host-ref="node1.example.com"> </configuration-service> <log-service id="LGS1" host-ref="node1.example.com"> </log-service> <administration-gui id="ADM1" host-ref="node1.example.com"> </administration-gui> <vespa id="VSP1" host-ref="node1.example.com"> </vespa> <storage-service id="STOR1" host-ref="node1.example.com" admin="root" adminpassword="d4tAs34rch" user="fast" userpassword="fast"> </storage-service> <name-service id="NAME1" host-ref="node1.example.com"> </name-service> </administration-set> </pre>	<p>Defining the administration node services.</p> <p>You should normally not need to modify this section.</p>
<pre> <data-source-set> <crawler id="CR1" host-ref="node1.example.com"> <property name="organization" value="Example"/> <property name="admin-email" value="admin@example.com"/> </crawler> </data-source-set> </pre>	<p>Defining the crawler service (if applicable).</p> <p>You should normally not need to modify this section.</p>
<pre> <content-distributor-set> <content-distributor id="CD1" host-ref="node1.example.com"/> </content-distributor-set> </pre>	<p>Defining the content distributor service.</p> <p>You should normally not need to modify this section.</p>

Table 6-1 Install Profile Example

Install Profile Elements/Attributes	Description
<pre> <search-engine-set> <search-engine id="RTS1" host-ref="node1.example.com"> <property name="search" value="true"/> <property name="index" value="true"/> </search-engine> <search-engine id="RTS2" host-ref="node2.example.com"> <property name="search" value="true"/> <property name="index" value="true"/> </search-engine> </search-engine-set> </pre>	<p>Defining the two search engine services. In this case the two nodes are combined search/index nodes.</p> <p>id: Any unique text string used as an ID for the search node.</p> <p>search="true" indicates that this is a search node.</p> <p>index="true" indicates that this is an indexer node.</p>
<pre> <document-processor-set> <document-processor id="DP1" host-ref="node1.example.com"/> </document-processor-set> </pre>	<p>Defining the document processing service.</p> <p>You should normally not need to modify this section.</p>
<pre> <query-result-processor-set> <query-result-processor id="QRP1" host-ref="node1.example.com"> <property name="default-language" value="en"/> <property name="languages" value="en"/> </query-result-processor> </query-result-processor-set> </pre>	<p>Defining the query & result processor service.</p> <p>default-language: Default language for linguistics.</p> <p>languages: Dictionaries are installed for these languages.</p>
<pre> <anchor-service> <anchor-server id="anch1" host-ref="node1.example.com"> <property name="dispatcher" value="true"/> </anchor-server> </anchor-service> </pre>	<p>Defining the Anchor Server (for authority / anchor text boosting) service.</p> <p>You should normally not need to modify this section.</p>
<pre> <status-service> <status-server id="STATUS1" host-ref="node2.example.com"> </status-server> </status-service> </pre>	<p>Defining the Status Server service.</p> <p>You should normally not need to modify this section.</p>
<pre> <license-services> <license-service id="LCS1" host-ref="node1.example.com"> </license-service> </license-services> </pre>	<p>Defining the License Manager service.</p> <p>You should normally not need to modify this section.</p>

Table 6-1 Install Profile Example

Install Profile Elements/Attributes	Description
<pre><search-engine-cluster-set> <search-engine-cluster id="cluster1"> <property name="use-memsearch" value="false"/> <property name="index-profile-type" value="standard"/> <search-engine-column id="col0"> <search-engine id-ref="RTS1"/> <search-engine id-ref="RTS2"/> </search-engine-column> <query-result-processors> <query-result-processor id-ref="QRP1"/> </query-result-processors> </search-engine-cluster> </search-engine-cluster-set></pre>	Defining the Search Engine Cluster. Ensure that the Search Engine nodes and Query and Result Engine nodes are listed here.
<pre></system-configuration></pre>	

Node Installation Procedure

Prerequisites

The following procedures requires that you have created or backed up an existing installation profile that was created during installation of FAST InStream.

The Installation profile can be found on the Admin Node of your installation:

```
$FASTSEARCH/InstallProfile.xml (UNIX)
%FASTSEARCH%\InstallProfile.xml (Windows)
```

Make sure that you have a proper backup of the Admin Node of your system. Refer to Chapter 3 *Backing Up and Restoring FAST InStream*.

Installing a Node According to the Install Profile

The following procedure can be applied if a node has to be (re)installed from scratch. If more than a single node has to be (re)installed, this procedure must be repeated for each of the nodes. The procedure includes:

- 1 If needed, (re)install the operating system and environment.
- 2 Copy the FAST InStream distribution to the new node.
- 3 Run the command-line installer program with custom options (see separate UNIX and Windows sections below).

Single Node Install on UNIX

The name of the binary will vary between UNIX platforms. This example is from Linux:

```
> setup_Linux.bin
-is:tempdir $TMP -silent
-G replaceNewerResponse=yesToAll
-P datasearch.installLocation=$FASTSEARCH
-W log.logOutput=<log file>
-W platform.resolveJVMExe=True
-W platform.resolveInstallLocation=False
-W stage.stage=remoteinstall
-W configparser.configSource=<path to InstallProfile.xml>
-W savecustomindexprofiles.source=<location of the index profile(s)>
-W platform.exitOnError=true
-W platform.exitOnWarning=true
```

See section *Installer Option Details* for documentation of the different options.

Single Node Install on Windows

```
> setup.exe -is:tempdir %TMP% -silent
-G replaceNewerResponse=yesToAll
-P datasearch.installLocation=%FASTSEARCH%
-W log.logOutput=<log file>
-W platform.resolveJVMExe=True
-W platform.resolveInstallLocation=False
-W stage.stage=remoteinstall
-W configparser.configSource=<path to InstallProfile.xml>
-W savecustomindexprofiles.source=<location of the index profile(s)>
-W password.username=<username>
-W password.password=<password>
-W platform.exitOnError=true
-W platform.exitOnWarning=true
```

See section *Installer Option Details* for documentation of the different options.

Example of a Single Node Install (Windows)

```
> setup.exe
-is:tempdir c:\fast-temp -silent
-G replaceNewerResponse=yesToAll
-P datasearch.installLocation=c:\fastsearch
-W log.logOutput=c:\fastsearch\log.txt
-W platform.resolveJVMExe=True
-W platform.resolveInstallLocation=False
-W stage.stage=remoteinstall
-W configparser.configSource=c:\fastsearch\InstallProfile.xml
-W savecustomindexprofiles.source=c:\fastsearch
-W password.username=MYDOMAIN\myusername
-W password.password=mypassword
-W platform.exitOnError=true
-W platform.exitOnWarning=true
```

Installer Option Details

The following table lists the options associated with the command-line installer program.

Table 6-2 Installer Options

Option	Description
-is:tempdir \$TMP	Set the temporary directory used by the installer.
-silent	Silent installation. No feedback/user interaction.
-G replaceNewerResponse=yesToAll	Automatically answer "yes" to questions.
-P datasearch.installLocation=\$FASTSEARCH	Set the installation directory. Note! This directory must be empty.
-W log.logOutput=<log file>	Set the log file used by the installer.
-W platform.resolveJVMExe=True	Make the installer resolve the path to the java JRE in use.
-W platform.resolveInstallLocation=False	Tell the installer not to guess the installation directory. Use the one specified by "-P datasearch.installLocation=\$FASTSEARCH"
-W stage.stage=remoteinstall	Make the installer only install on this node, even if the install profile describes a multi node installation.
-W configparser.configSource=<path to Install-Profile.xml>	Use the specified install profile.
-W savecustomindexprofiles.source=<location of any custom index profiles>	Take any custom index profiles from the specified directory.
-W password.username=<username>	Username, needed when installing services on windows.
-W password.password=<password>	Password, needed when installing services on windows.
-W platform.exitOnError=true	Exit at once if an error has occurred.
-W platform.exitOnWarning=true	Exit at once if a warning has occurred.

Configuring new Nodes in the Install Profile

Configuring a new Search/Index Node

To configure a new search/index node in the install profile, edit the *InstallProfile.xml* file located in *\$FASTSEARCH/* (Unix) or *%FASTSEARCH%* (Windows) in the following way. The procedure applies both when adding a new column or adding a new row:

- 1 Add the new node in the `<host-set>` section. Copy the existing `<host>` entry and change the 'id' and 'hostname' to reflect the fully qualified hostname for the new host. Example:

```
<host-set>
  <host id="node1.example.com">
    <property name="hostname" value="node1.example.com"/>
    <property name="platform" value="Linux"/>
    <property name="username" value="username"/>
    <property name="install-dir" value="/home/username/fast"/>
    <property name="temp-dir" value="/tmp"/>
    <property name="remote-cmd" value="SSH2"/>
    <property name="baseport" value="13000"/>
  </host>
  <host id="node2.example.com">
    <property name="hostname" value="node2.example.com"/>
    <property name="platform" value="Linux"/>
    <property name="username" value="username"/>
    <property name="install-dir" value="/home/username/fast"/>
    <property name="temp-dir" value="/tmp"/>
    <property name="remote-cmd" value="SSH2"/>
    <property name="baseport" value="13000"/>
  </host>
</host-set>
```

- 2 Add the new node in the `<search-engine-set>` section. Copy an existing `<search-engine>` entry and change the 'id' and 'hostname'.

```
<search-engine-set>
  <search-engine id="RTS3" host-ref="node1.example.com">
    <property name="search" value="true"/>
    <property name="index" value="true"/>
  </search-engine>
  <search-engine id="RTS4" host-ref="node2.example.com">
    <property name="search" value="true"/>
    <property name="index" value="true"/>
  </search-engine>
</search-engine-set>
```

- a If you add a combined search/index node the properties "search" and "index" should be set to "true".
- b If you add a dedicated indexer node the property "search" should be set to "false".

- c** If you add a dedicated search node the property “index” should be set to “false”.

Note! If you add a column make sure that the new column has the same number of search nodes and indexers as the existing columns, e.g. both “search” and “index” need to be set to “true” for the same number of nodes.

- 3** Add the new node in the <search-engine-cluster> section.

- a** Adding a new column:

Copy an existing <search-engine-column> entry and change the ‘id’ and ‘id-ref’.

```
<search-engine-cluster>
  <property name="use-memsearch" value="false"/>
  <property name="index-profile-type" value="standard"/>
  <search-engine-column id="col0">
    <search-engine id-ref="RTS3"/>
  </search-engine-column>
  <search-engine-column id="col1">
    <search-engine id-ref="RTS4"/>
  </search-engine-column>
  <query-result-processors>
    <query-result-processor id-ref="QRP1"/>
  </query-result-processors>
</search-engine-cluster>
```

- b** Adding a new row:

Copy an existing <search-engine> entry and change the ‘id-ref’.

```
<search-engine-cluster id="cluster1">
  <property name="use-memsearch" value="false"/>
  <property name="index-profile-type" value="standard"/>
  <search-engine-column id="col0">
    <search-engine id-ref="RTS1"/>
    <search-engine id-ref="RTS2"/>
  </search-engine-column>
  <query-result-processors>
    <query-result-processor id-ref="QRP1"/>
  </query-result-processors>
</search-engine-cluster>
```

- 4** Save the *InstallProfile.xml* file.

Chapter 7

Adding a new Node using the OEM Installer

About this Chapter

This chapter provides common steps in order to add a new node to a FAST InStream installation using the FAST InStream OEM Installer. The overall procedure for the different scenarios is described in Chapter 5 *Applying Run-Time Deployment Changes*:

- *Adding a Combined Index and Search Row on page 44*
- *Adding a Dedicated Search Row on page 46*
- *Adding a new Query and Result Engine on page 48*
- *Adding a Combined Index and Search Column on page 50*

Convert a single-node installation into a node in an existing multi-node installation

This section assumes that you have installed the new node as a single-node installation, typically using the FAST InStream OEM Installer.

In the following (a) indicates the Admin Node in the existing FAST InStream installation, and (n) indicates the new node to be added.

Installing a Combined Search/indexer Node (new row or column)

The sub-steps needed are described below:

- 1 Ensure that FAST InStream is not started on the newly installed node.
- 2 Change \$FASTSEARCH/etc/CSLocation.xml on node (n) to point to the Admin Node (a).
- 3 Edit \$FASTSEARCH/etc/NodeConf.xml:
 - a Remove *nameservice*, *httpd*, *lmgrd*, *logserver*, *configserver*, *contentdistributor*, *cachemanager*, *statusserver*, *anchorserver-storage*, *storageservice*, *j2ee* from the `<startorder>` element.
 - b *qrserver* should also be removed unless the new node also includes a Query & Result Engine (QRServer).
 - c Change the nameservice reference to procserver's command line arguments to point to the already existing admin node:

```
<parameters>-b 1 -O -c 0 -P $PORT -ORBInitRef
NameService=corbaname::node02.yourdomain.com:16099</parameters>
```
- 4 Edit \$FASTSEARCH/etc/NodeState.xml:
 - Remove all `<process>` entries from `<dynamic>`
- 5 Edit etc/LoggerConfig.xml:
 - Specify the hostname of the Admin node in `server->hostname`
- 6 Enable search-1 process on new node by editing etc/searchrc-1.xml:
 - edit namingservice host to the hostname of the Admin node
 - Edit columnrowid to real value ("0-1" in this example)

Installing a Dedicated Search Node

The sub-steps needed are described below:

- 1 Ensure that FAST InStream is not started on the newly installed node.

- 2 Change \$FASTSEARCH/etc/CSLocation.xml on node (n) to point to the Admin Node (a).
- 3 Edit \$FASTSEARCH/etc/NodeConf.xml:
 - a Remove *nameservice*, *httpd*, *lmgrd*, *logserver*, *configserver*, *contentdistributor*, *cachemanager*, *statusserver*, *anchorserver-storage*, *storageservice*, *j2ee* from the `<startorder>` element.
 - b *qrservice* should also be removed unless the new node also includes a Query & Result Engine (QRServer).
 - c Change the *nameservice* reference to *proccserver*'s command line arguments to point to the already existing admin node:

```
<parameters>-b 1 -O -c 0 -P $PORT -ORBInitRef
NameService=corbaname::node02.yourdomain.com:16099</parameters>
```
- 4 Edit \$FASTSEARCH/etc/NodeState.xml:
 - Remove all `<process>` entries from `<dynamic>`
- 5 Edit etc/LoggerConfig.xml:
 - Specify the hostname of the Admin node in `server->hostname`
- 6 Disable indexer process on new node (remove from `<startorder>` list in \$FASTSEARCH/NodeConf.xml)
- 7 Enable search-1 process on new node by editing etc/searchrc-1.xml:
 - edit *namingservice* host to the hostname of the Admin node
 - Edit *columnrowid* to real value ("0-1" in this example)
 - Set *copyindex* to true.
- 8 Start the new node
- 9 Disable search-1 process on original node (remove from `<startorder>` list in NodeConf.xml)
- 10 Restart *topfdispatch*

Chapter 8

Changing the Host Name of Your Installation

About this Chapter

This chapter describes how to change the host name in your FAST InStream installation.

It includes:

- Overview
- Changing the Host Name in a UNIX System
- Changing the Host Name in a Windows System

Note! It is not recommended to change the host name of a running installation unless strictly necessary.

Overview

Changing the host name of your FAST InStream installation consists of the following overall steps:

- 1 Stopping FAST InStream.
- 2 Retrieving all involved files in the installation directory.
- 3 Replacing the old with the new host name in all retrieved files.
- 4 Cleaning the installation directory for old name service files.
- 5 Repeating steps 2 through 4 for all nodes in your system.
- 6 Restarting FAST InStream.

These steps are detailed in platform specific procedures listed below.

Note! All procedures are based on the assumption that both the old and the new host name are fully qualified host names.

Changing the Host Name in a UNIX System

To change the host name used in FAST InStream on a UNIX system, follow the procedure below:

- 1 Stop FAST InStream using a sequential shutdown. For details, refer to Chapter 2 *Starting and Stopping FAST InStream*, section *Running a Sequential Shutdown*.
- 2 Perform a search (grep) in all FAST InStream directories for files that contain the host name.

Note! It is recommended that you take a backup copy of all files before changing them.

- 3 Replace the old host name with the new one in each of the files.

If Perl is available on your system, you may for example use the following Perl command:

```
find etc -type f -print | xargs perl -pi.bak -e 's/
<oldname>\.<old>\.domain/<newname>\.<new>\.domain/gi'
```

with <oldname>\.<old> representing the old host name and <newname>\.<new> representing the new host name.

This command generates a backup copy (*.bak) of all files in the *etc* directory, replaces the old host name with the new in each of the files containing the old host name and saves the files.

- 4 Delete the contents of the following directories:

var/etc

var/log/omniorb

- 5 Repeat steps 2 through 4 on all other nodes in your system.
- 6 Restart FAST InStream. For details, refer to Chapter 2 *Starting and Stopping FAST InStream*.

Changing the Host Name in a Windows System

To change the host name used in FAST InStream on a Windows system, follow the procedure below:

- 1 Stop FAST InStream using a sequential shutdown. For details, refer to Chapter 2 *Starting and Stopping FAST InStream*, section *Running a Sequential Shutdown*.
- 2 Perform a Search in all FAST InStream directories for files that contain the host names.

Note! It is recommended that you take a backup copy of all files before changing them.

- 3 Replace the old host name with the new one in each of the files.
- 4 Delete the contents of the following directories:
 `var\etc`
 `var\log\omniorb`
- 5 Repeat steps 2 through 4 on all other nodes in your system.
- 6 Restart FAST InStream. For details, refer to Chapter 2 *Starting and Stopping FAST InStream*.

Chapter 9

Gathering Support Information

About this Chapter

This chapter describes which type of information you need to submit to FAST Support and how you gather this information.

It includes:

- Information Gathering
- Retrieving Product Version Information
- Retrieving Platform/Operating Systems Information
- Overall System Diagnosis Information
- Product Status Information
- Log File Information

Information Gathering

Before you contact FAST Support with an issue, gathering the following information will help to identify and solve the issue in a timely manner. Provide the following:

- the version of product you are running
- the platform/operating system on which you are running
- the type of system, production or development, that is encountering the issue to help determine the urgency of the issue
- a detailed description of the issue
- any necessary log files for background information
- detailed steps to reproduce the issue

The information provided in the

The following sections explain how to obtain this information.

Retrieving Product Version Information

Use any of the following actions to locate the version of product that you are running:

- Check the label of your FAST InStream distribution CD
- Check the latest upgrade downloaded from *ftp.fast.no*
- Click on **System Overview** in the administrator interface and note the **Build-number**

Retrieving Platform/Operating Systems Information

To obtain the existing operating system information:

For **Windows**, verify whether you are running on Microsoft Windows 2000 or 2003.

For **UNIX**, log on as a UNIX user and type in the following commands: `uname` and `uname -a`

Overall System Diagnosis Information

In addition to version levels, the following usage and configuration information is helpful in describing details about the issue:

- Environment Variables
- License Status
- Machine Load and Memory
- Disk Space and CPU Usage
- IP/Network Addresses
- Ports Allocation

Environment Variables

This section describes system and machine environment variables.

System Environment

- The *FASTSEARCH* variable must point to the root of the FAST InStream installation directory.
- *LM_LICENSE_FILE* variable must point to the installed license file, *\$FASTSEARCH/etc/fastsearch.lic*.
- *LD_LIBRARY_PATH* (Solaris/Linux)/*LIBPATH* (AIX)/*SHLIB_PATH* (HPUX) environment variable provides the location of the shared libraries and must point to *\$FASTSEARCH/lib*.

Refer to the *FAST InStream Installation Guide*, *Installation Requirements and Concepts* chapter, and the *Setup for FAST InStream in a UNIX/Windows Environment* chapters for more information.

Machine Environment

For **Windows**, right-click on **My Computer** and select **Properties**. Information for the system is listed under the **General** tab.

For **UNIX (Linux, Solaris, or HP-UX)**, type in the following commands: `uname -a` and `ulimit -a`

For **UNIX (AIX)**, type in the following commands: `uname -x` and `ulimit -a`

License Status

If you encounter issues with the license file:

- Check that the host name of the License Manager is correct in the **SERVER** line of the license file. The license file can be found in the *\$FASTSEARCH/etc* (for UNIX) and *%FASTSEARCH\etc* (for Windows) directory of the server running the License Manager (mostly the Admin node)
- Check to make sure that the license has not expired (if applicable). This should be described in the header text of the license file

If these suggestions do not solve the issue, then go to the following files and report your findings:

For **Windows**:

%FASTSEARCH%\etc\fastsearch.lic

%FASTSEARCH%\var\log\lmgrd.scrap (log file for License Manager)

all.log files in the *InstallProfile.xml* file

For **UNIX**:

\$FASTSEARCH/etc/fastsearch.lic

\$FASTSEARCH/var/log/lmgrd.scrap (log file for License Manager)

all.log files in the *InstallProfile.xml* file

Machine Load and Memory

For **Windows**, run **msinfo32** and provide the **System Summary** information.

For **UNIX (Linux)**, use the **free** and **vmstat** commands to get memory statistics.

For **UNIX (Solaris, AIX or HP-UX)**, use the **vmstat** command to get memory statistics.

Disk Space and CPU Usage

For **Windows**, go into **Task Manager** and select the **Processes** tab to view CPU and memory usage.

For **UNIX**, use the **iostat** and **top** commands.

IP/Network Addresses

For **Windows**, use the `ipconfig` command.

For **UNIX**,

- Check the `hostname` and `/etc/hosts` and `/etc/resolv.conf` files.
- Use the `ifconfig -a` command.
- Use the `nslookup` command for DNS check.

Ports Allocation

Use the **System Overview** selection in the FAST InStream administrator interface to view the main ports used by the FAST InStream modules. For more detailed information:

For **Windows**, use the `netstat` command.

For **UNIX**, use the `netstat -p` command.

Product Status Information

This section describes how to obtain process and documents status in FAST InStream.

Process Status in Process Window

Use the and **System Overview** selections in the FAST InStream administrator interface to view process status. For more detailed information:

For **Windows** (OS level): Go into **Task Manager** and select the **Processes** tab to view the status of all processes.

For **UNIX** (OS level): Use the `ps` command.

For **FAST InStream**: Use the `./nctrl sysstatus` command.

Documents Status

If documents are not being indexed but have been crawled, it is important to determine where in the process the document gets dropped. It is possible to trace a document to see how it has been processed. Complete the following procedure to trace a document's progress.

For **Windows**:

- 1 Execute `%FASTSEARCH%\bin\psctrl doctrace on` from the command prompt. This utility enables document tracing as documents flow through the pipeline.
- 2 Add a single document that contains the data in question to a collection.
- 3 Execute `%FASTSEARCH%\bin\doclog -a` to monitor how the document is being modified by the pipeline.
- 4 Check the log files for any possible errors. In such cases, there will be a mismatch between the number of documents in the Content Distributor (accessed from the **Collection Overview** selection) and the number of documents in the Index.

For **UNIX**:

- 1 Execute `$FASTSEARCH/bin/psctrl doctrace on` from the command prompt. This utility enables document tracing as documents flow through the pipeline.
- 2 Add a single document that contains the data in question to a collection.
- 3 Execute `$FASTSEARCH/bin/doclog -a` to monitor how the document is being modified by the pipeline.
- 4 Check the log files for any possible errors. In such cases, there will be a mismatch between the number of documents in the Content Distributor (accessed from the **Collection Overview** selection) and the number of documents in the Index.

Log File Information

This section provides log file information that can help isolate an issue.

Search Node (Indexing) issues

Log files:

RTSearch log files:

var/log/rtsearch/.log*

RTSearch configuration files:

etc/config_data/RTSearch/webcluster/rtsearchrc.xml,

etc/config_data/RTSearch/webcluster/index-profile

General system log:

var/log/all.log

Query and Result Processing issues

Log files:

var/log/all.log

var/log/qserver.scrap

var/log/searchctrl/.**

Action:

- 1 Send exact product description with snapshot of the **System Management** pages.
- 2 Include information on any changes (network or FAST InStream configuration) that might have contributed to the issue.
- 3 Send a snapshot of the search view including error messages.

Crawler issues

Log files:

Crawler log files:

*var/log/crawler/**

Crawler configuration files:

bin/adminclient -G <collection name> > collname.xml

Collection statistics:

bin/adminclient -q <collection name> > collstat.xml

Note that you may have used a different name than the default.

Configuration Server issues

Log files:

From the Configuration Server:

var/log/configserver.scrap

var/log/.log*

Action:

Check the *var/log/archive/* folder for the latest zipped file. If the Error occurred before the current log, then attach the latest zipped *all.log*.

Document Processing issues

Log files:

var/log/procserver_x.scrap

var/log/.log*

Action:

- 1** From the Configuration Server, run the tool *bin/psctrl status* in order to check the process status.
 - 2** In case of missing documents, see *Documents Status* on page 86.
 - 3** Send the procserver scrap and logs (*var/log/*.log*).
-

File Traverser issues

Log files:

Enable filetraverser logging with the *-l* parameter

var/log/all.log

Action:

- 1** Send the exact command or batch files being used for content feeding.
 - 2** Cut and paste the output from the filetraverser.
 - 3** Send exact problem description in snapshot or log files.
-

License Manager issues

Log files:

etc/fastsearch.lic
var/log/lmgrd.scrap
var/log/all.log

Action:

If components are not coming up correctly or not coming up at all:

- 1 Send the *etc/fastsearch.lic* file.
- 2 Include *var/log/lmgrd.scrap* and *var/log/all.log*.
- 3 Send any network configuration change information.

If a Query and Result Processing limitation appears:

- 1 Send a snapshot of *http://<search nodes>:15101/status/*.
 - 2 Send the *etc/fastsearch.lic* file.
 - 3 Include *var/log/lmgrd.scrap* and *var/log/all.log*.
-

Status Server issues

Log files:

var/log/statusserver.scrap
*var/log/StatusServer\@*_0.log files*
var/log/all.log file

Action:

- 1 Send *var/log/statusserver.scrap* and *var/log/StatusServer\@*_0.log* files.
 - 2 Send *var/log/all.log* file.
-

Anchor Server issue

Log files:

var/log/anchorserver-storage.scrap
var/log/AnchorServer\@.log*
var/log/all.log

Action:

- 1 Send *var/log/anchorserver-storage.scrap* and *var/log/AnchorServer\@*.log* files.
 - 2 Send *var/log/all.log* file.
-

Content API or SDK issue

Log files:

var/log/all.log

var/log/ Content\ Distributor\@.log*

Action:

- 1 Send exact snippet messages from the feeder.
 - 2 Send out the code itself.
 - 3 Send the *var/log/all.log* and *var/log/ Content\ Distributor\@*.log*.
-

Business Manager's Control Panel issues

Log files:

var/log/j2ee.scrap

var/log/all.log

var/log/querylogs/< query_log.date_queries_not_found.log>

Action:

If it is a Rank Tuning or document Boosting issue from the interface:

- 1 Send exact description of the problem in snapshots.
- 2 Send *var/log/j2ee.scrap* and *var/log/all.log* files.
- 3 Send exact problem description.

If it is a Query Reporting issue:

- 1 Send *var/log/j2ee.scrap* and *var/log/all.log* files.
- 2 Send *var/log/querylogs/< query_log.date_queries_not_found.log>*.
- 3 Send snapshot Snapshot of the *http://<Admin Server>:16089/bmcp/*.

If it is an internal connection issue:

- 1 Send a snapshot of the *http://<Admin Server>:16089/bmcp/*.
 - 2 Send the *var/log/j2ee.scrap* and *var/log/all.log* files.
 - 3 Send snippet of *bin/nctrl sysstatus*.
-

Index

A

anchor server
 backing up 28
 restoring 28
authority ranking 14

B

BMCP rank tuning 13

C

collection specific configuration 20
content
 ... dynamics 42
 ... volume 42
crawler
 suspending 6
crawler configuration
 backing up 18
 re-feeding 19
 restoring 19

D

dictionaries
 backing up 26
 restoring 26
document processing
 suspending 7

E

environment variables

machine 83
system 83
unix 2
windows 2

F

filetraverser
 suspending 7
fixml 14, 29

H

hostname, changing 77

I

index profile 12
indexing
 backing up 29
 resuming 10
 suspending 8
install profile 16
 InstallProfile.xml 12
installer
 options 70

Q

query rate 42

R

rank tuning
 backing up 24

- redeploying 24
- restoring 24

S

- shutdown

- sequential 5
 - simple, unix 9
 - simple, windows 9

- single node install

- example 69
 - UNIX 68
 - windows 69

- starting fast

- unix 4
 - windows 4

- status server 13

- backing up 22
 - recovering 22

- support

- disk space and cpu usage 84
 - gathering information 82
 - ip/network addresses 85
 - license status 84
 - log file information 87
 - machine load and memory 84
 - platform/operating system 82
 - ports allocation 85
 - product status
 - documents status 86
 - process status 85
 - product version 82

- system configuration 12

- collection specific 12
 - document processing 13
 - install profile 12