

# [MS-FSICFG]: Indexer Configuration File Format

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
02/19/2010	1.0	Major	Initial Availability
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Minor	Updated the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.5	Minor	Clarified the meaning of the technical content.
04/11/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>4</b>
1.1 Glossary	4
1.2 References	4
1.2.1 Normative References	4
1.2.2 Informative References	5
1.3 Structure Overview (Synopsis)	5
1.3.1 Options	6
1.3.2 Network	7
1.3.3 Index scheduling	7
1.3.4 Post processors	7
1.3.5 Fault-tolerance	7
1.4 Relationship to Protocols and Other Structures	8
1.5 Applicability Statement	8
1.6 Versioning and Localization	8
1.7 Vendor-Extensible Fields	8
<b>2 Structures</b>	<b>9</b>
2.1 Global elements	9
2.2 Complex Types	9
2.2.1 CT_options	9
2.2.2 CT_network	11
2.2.3 CT_index-scheduling	12
2.2.4 CT_post-processors	13
2.2.5 CT_processor	13
2.2.6 CT_ft	13
2.2.7 CT_real-time-properties	14
<b>3 Structure Examples</b>	<b>15</b>
<b>4 Security Considerations</b>	<b>16</b>
<b>5 Appendix A: Full XML Schema</b>	<b>17</b>
<b>6 Appendix B: Product Behavior</b>	<b>19</b>
<b>7 Change tracking</b>	<b>20</b>
<b>8 Index</b>	<b>21</b>

# 1 Introduction

This document specifies the Indexer Configuration File Format, an XML-based configuration format for configuring the indexing nodes in an enterprise search service. This format describes configuration parameters for changing general behavior and timing intervals, and for changing parameters for performance tuning.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**fault-tolerant**

The following terms are defined in [\[MS-OFCGLOS\]](#):

**base port**  
**exclusion list**  
**FAST Index Markup Language (FIXML)**  
**index column**  
**index partition**  
**indexer row**  
**indexing component**  
**indexing node**  
**item**  
**master indexer node**  
**query matching component**  
**query matching node**  
**Web crawler**

The following terms are specific to this document:

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-FSCX] Microsoft Corporation, "[Configuration \(XML-RPC\) Protocol Specification](#)".

[MS-FSIFT] Microsoft Corporation, "[Indexer Fault Tolerance Protocol Specification](#)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

[XMLSCHEMA] World Wide Web Consortium, "XML Schema", September 2005, <http://www.w3.org/2001/XMLSchema>

## 1.2.2 Informative References

[ISO/IEC-29500-4] International Organization for Standardization, "Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 4: Transitional Migration Features", ISO/IEC 29500-4:2008, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=51462](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51462)

[MS-FSIFXML] Microsoft Corporation, "[FIXML Data Structure](#)".

[MS-FSIPA] Microsoft Corporation, "[Index Publication and Activation Protocol Specification](#)".

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-OFGLGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

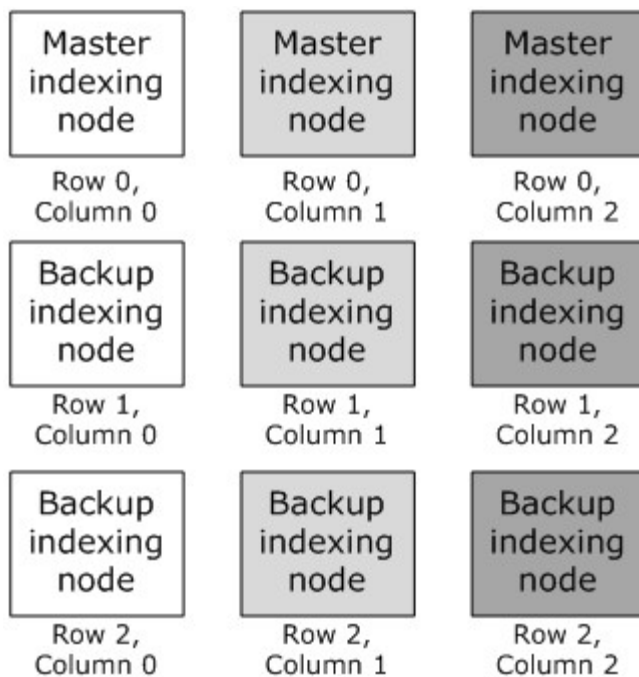
## 1.3 Structure Overview (Synopsis)

This document describes the Indexer Configuration File Format, an XML-based configuration format for configuring the **indexing nodes** in an enterprise search service. This format includes configuration parameters both for changing general behavior and timing intervals, as well as parameters for performance tuning.

**Item** operations, such as adding new items or removing old items, are sent to the indexing node in batches containing several operations. Due to load balancing at higher levels, items sent by a **Web crawler**, for example, can arrive at the indexer in a different order than that in which they were initially sent. Batches that arrive out of order are first placed on the **spare** queue. Batches that arrive in the correct order are placed on the **api** queue. After a batch has been placed on the api queue, the spare queue is checked for batches that could be placed on the api queue to form a contiguous range of batches. There is one spare queue for every client feeding the indexing node.

On the highest level, the total index is partitioned across several **index columns** of indexing nodes. On the level of each indexing node, the index is partitioned into a disjointed set of **index partitions**. These index partitions are denoted by integers from 0 to n-1, where n is the number of index partitions on the indexing node. A full set of disjoint index partitions is called an index set, and the set of index partitions currently used by **query matching nodes** to facilitate search queries is called the active index set.

In a **fault-tolerant** system setup there are several indexing nodes in the same index column, with indexing nodes assuming different **column roles**. In every index column, one of the indexing nodes assumes the role of the **master indexer node**, while the rest are referred to as backup indexing nodes. The different indexing nodes are identified by their **indexer row** identifier, as shown in the following figure.



**Figure 1: Matrix of indexing nodes**

Prior to indexing, items are stored in the intermediate **FAST Index Markup Language (FIXML)** format, as described in [\[MS-FSFIXML\]](#). A single FIXML structure contains one or more items. The validity of the items in a FIXML structure is decided by its corresponding FIXML meta structure.

The high level item operations are converted by the master indexer node into low level sequence operations, as described in [\[MS-FSIFT\]](#) section 2.2.2. Each indexing node keeps a fault-tolerance storage, which is a backlog of previously-processed sequence operations. The fault-tolerance storage enables a restarted backup indexing node to quickly become fully synchronized; the master indexer node is only required to send the sequence operations that were not delivered during the backup indexing node's downtime.

Query matching nodes register with the master indexer node using the protocol described in [\[MS-FSIPA\]](#). Some of the query matching node's configuration parameters are derived from the configuration format specified in this document.

The configuration parameters are grouped into different child elements of the root element. The child elements containing the actual parameters are described in the following sections.

### 1.3.1 Options

General options are grouped into a single child element, which specifies the following information:

- The encoding of the FIXML files.
- The sizes of the index partitions.
- The maximum number of items stored in each FIXML file.
- The maximum number of FIXML set directories allowed.

- The maximum number of FIXML files allowed.
- The minimum amount of disk space allowed before warnings are logged.
- The level of logging.
- The number of index partitions.
- The interval at which **exclusion lists** are sent to the query matching nodes.
- The number of index partitions allowed to re-index simultaneously.
- The way in which dictionary normalization is performed.
- The interval at which dictionary normalization takes place.
- The directory in which FIXML files are stored.
- The directory in which the inverted index structures are stored.
- The maximum size of the api queue.

### 1.3.2 Network

A single child element stores specified network options that include the following:

- The fully qualified host name of the indexing node.
- The **base port** of the indexing node.

### 1.3.3 Index scheduling

The index scheduling options are grouped into a single child element specifying several items, including the following:

- The type of index scheduling algorithm used.
- The parameters of the index scheduling algorithm, informing when an index partition is to be re-indexed.

### 1.3.4 Post processors

The post processors options are grouped into a single child element specifying items that include the following:

- The external processes that work on the index structures.

### 1.3.5 Fault-tolerance

The fault-tolerance options are grouped into a single child element specifying items that include the following:

- The enabling or disabling of fault-tolerance.
- The location of the fault-tolerance storage.
- The interval at which indexing nodes reassess their column role.

- The maximum amount of disk space used by the fault-tolerance storage.

#### **1.4 Relationship to Protocols and Other Structures**

The configuration file described in this file format specification is downloaded by the **indexing components** from the configuration service described in [\[MS-FSCX\]](#).

#### **1.5 Applicability Statement**

This file format can be used to configure indexing nodes in an enterprise search service.

#### **1.6 Versioning and Localization**

None.

#### **1.7 Vendor-Extensible Fields**

None.



## 2 Structures

The Indexer Configuration File Format is defined in XML schema, as specified in [\[XMLSCHEMA\]](#).

The indexing components retrieve the configuration structure specified in this document by calling the **LoadConfigFile** method, as specified in [\[MS-FSCX\]](#) section 2.2.26. The **LoadConfigFile** method utilizes the following parameters:

**module:** A string value that MUST be "RTSearch".

**filepath:** A string value that MUST be "webcluster/rtsearchrc.xml".

### 2.1 Global elements

The global element contains an indexing node's configuration parameters. An Indexer Configuration File structure MUST contain only one **configuration** element.

```
<xs:element name="configuration" type="CT_configuration" />
```

### 2.2 Complex Types

#### 2.2.1 CT\_options

*Referenced by:* CT\_configuration.

The **CT\_Options** complex type specifies the general configuration parameters of an indexing node, as follows.

```
<xs:complexType name="CT_options">
  <xs:attribute name="xmlEncoding" type="xs:string" use="required"/>
  <xs:attribute name="docsDistributionMax" type="xs:string" use="required"/>
  <xs:attribute name="numberDocsPerFixml" type="xs:int" use="required"/>
  <xs:attribute name="docsDistributionSteps" type="xs:int" use="required"/>
  <xs:attribute name="docsDistributionSamples" type="xs:int" use="required"/>
  <xs:attribute name="fsearchCachePstDist" type="xs:string" use="required"/>
  <xs:attribute name="maxMBCacheSize" type="xs:string" use="required"/>
  <xs:attribute name="maxSetDirs" type="xs:int" use="required"/>
  <xs:attribute name="maxFixmlFiles" type="xs:int" use="required"/>
  <xs:attribute name="fixmlPrefetchThreads" type="xs:int" use="required"/>
  <xs:attribute name="diskSpaceMBWarning" type="xs:int" use="required"/>
  <xs:attribute name="indexerSwapDetect" type="xs:string" use="required"/>
  <xs:attribute name="debugLog" type="xs:boolean" use="required"/>
  <xs:attribute name="numberPartitions" type="xs:int" use="required"/>
  <xs:attribute name="exclusionlistInterval" type="xs:int" use="required"/>
  <xs:attribute name="indexingThreads" type="xs:int" use="required"/>
  <xs:attribute name="maxActiveIndexingJobs" type="xs:int" use="required"/>
  <xs:attribute name="removeCollectionBatchSize" type="xs:int" use="required"/>
  <xs:attribute name="normalizeMode" type="xs:string" use="required"/>
  <xs:attribute name="normalizeInterval" type="xs:int" use="required"/>
  <xs:attribute name="fixmlPath" type="xs:string" use="required"/>
  <xs:attribute name="indexDir" type="xs:string" use="required"/>
  <xs:attribute name="maxQueueSize" type="xs:int" use="required"/>
  <xs:attribute name="maxMemPerDocIndex" type="xs:int" use="required"/>
</xs:complexType>
```

**Child elements:** None.

**Attributes**

Name	Description
<b>xmlEncoding</b>	A string value specifying the value to be inserted into the <b>encoding</b> attribute of the FIXML files. If the length of the string is less than or equal to 1, the <b>encoding</b> attribute MUST NOT be inserted.
<b>docsDistributionMax</b>	A comma-separated list of integers specifying the upper limit of allowed items in an index partition. The values are given in increasing index partition order, where the first integer value corresponds to index partition 0.
<b>numberDocsPerFixml</b>	An integer value specifying the maximum number of items per FIXML file.
<b>docsDistributionSteps</b>	An integer value specifying implementation specific distribution data, which SHOULD be 100.
<b>docsDistributionSamples</b>	An integer value specifying implementation specific distribution data, which SHOULD be 100.
<b>fsearchCachePstDist</b>	A comma-separated list of integers specifying implementation specific cache distribution percentages, which SHOULD be "5,20,7,20,0,5,3,0,4,5,11,0,20".
<b>maxMBCacheSize</b>	This value may be specified in two ways: An integer value specifying the maximum memory usage, in megabytes, of the <b>query matching component</b> . An integer value specifying the percentage of total physical memory that is allowed for use by the query matching component, followed by a percentage sign.
<b>maxSetDirs</b>	An integer value specifying the maximum number of FIXML set directories allowed. FIXML files are identified by increasing sequence identifiers, starting at 0. These sequence identifiers are then mapped to filenames and directories using the following algorithm:  <code>FILENAME = sequence identifier % 250</code> <code>DIRECTORY = sequence identifier / 250</code>
<b>maxFixmlFiles</b>	An integer value specifying the maximum number of FIXML files allowed.
<b>fixmlPrefetchThreads</b>	An integer value specifying an implementation specific number of threads, which SHOULD be 20.
<b>diskSpaceMBWarning</b>	An integer value specifying the smallest amount of disk space, in megabytes, allowed before warnings will be logged. Warnings MUST be logged if the available disk space is less than the configuration value for either of the following folders: The directory containing the fault-tolerance storage. The directory containing the inverted index structures. The directory containing the FIXML files.
<b>indexerSwapDetect</b>	A string value specifying the sizes of implementation specific memory structures, which SHOULD be "-k 600 -K 1200".
<b>debugLog</b>	A Boolean value specifying whether or not debug level logging is enabled.

Name	Description
<b>numberPartitions</b>	An integer value specifying the number of index partitions on the indexing node.
<b>exclusionlistInterval</b>	An integer value specifying the number of seconds that MUST pass between the time exclusion list updates that are sent to the query matching nodes.
<b>indexingThreads</b>	An integer value specifying the number of implementation specific threads, which SHOULD be 4.
<b>maxActiveIndexingJobs</b>	An integer value specifying the maximum number of index partitions that are allowed to re-index simultaneously.
<b>removeCollectionBatchSize</b>	An integer value specifying implementation specific batch sizes, which SHOULD be 5000.
<b>normalizeMode</b>	A string value specifying when index normalization occurs. Index normalization is a scheme to normalize individual index partitions so that their term frequencies are relative to the entire index set, rather than the individual index partitions. The value MUST be either "synchronized" or "periodic". If the parameter is set to "synchronized", the normalization MUST occur directly after an index partition has been re-indexed. If the parameter is set to "periodic", the index partitions MUST be rank normalized at set time intervals, as configured using <b>normalizeInterval</b> .
<b>normalizeInterval</b>	An integer value specifying the number of seconds that MUST pass between index normalizations. This value is only valid if <b>normalizeMode</b> is set to "periodic".
<b>fixmlPath</b>	A string value specifying the directory in which the FIXML files are stored.
<b>indexDir</b>	A string value specifying the directory in which the inverted index structures are stored.
<b>maxQueueSize</b>	An integer value specifying the maximum size, in bytes, of the api queue and the spare queue.
<b>maxMemPerDocIndex</b>	An integer value specifying implementation specific memory management data, which SHOULD be 500.

## 2.2.2 CT\_network

Referenced by: CT\_network.

The **CT\_network** complex type specifies the network-related configuration parameters of an indexing node, as follows.

```
<xs:complexType name="CT_network">
  <xs:attribute name="qualifiedHostName" type="xs:string" nullable="true"/>
  <xs:attribute name="useStrictBind" type="xs:string" use="required"/>
  <xs:attribute name="basePort" type="xs:string" use="required"/>
</xs:complexType>
```

**Child elements:** None.

**Attributes**

Name	Description
<b>qualifiedHostName</b>	A string value specifying the fully qualified domain name of the indexing node. If the attribute is empty or missing, the host name will be automatically resolved.
<b>useStrictBind</b>	Implementation specific configuration information.
<b>basePort</b>	A string specifying the base port of the indexing service.

### 2.2.3 CT\_index-scheduling

Referenced by: CT\_index-scheduling.

The **CT\_index-scheduling** complex type specifies the index scheduling scheme of the indexing node, as follows.

```
<xs:complexType name="index-scheduling">
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="triggers" type="xs:string" use="required"/>
</xs:complexType>
```

**Child elements:** None.

#### Attributes

Name	Description
<b>type</b>	A string specifying the index-scheduling algorithm used. The supported algorithm is: <ul style="list-style-type: none"> <li>▪ <b>docCountArchive</b></li> </ul>
<b>triggers</b>	<p>A comma-separated list of integers specifying the thresholds of the individual index partitions. When the number of items in index partition <math>n-1</math> is greater than the threshold for index partition <math>n</math>, index partition <math>n</math> will be scheduled for re-indexing.</p> <p>The values are given in increasing index partition order, the first integer value corresponding to index partition 1. Index partition 0 does not have a trigger value, as it is automatically triggered as soon as a new item has been added.</p> <p>If the <b>triggers</b> list contains less than <math>n-1</math> values, where <math>n</math> is the number of index partitions, the thresholds of the remaining index partitions will be deduced from the <b>docsDistributionMax</b> attribute of the <b>options</b> child element. If an index partition <math>p</math> does not have a threshold value specified in <b>triggers</b>, it will be scheduled for re-indexing when the total number of items indexed in index partitions 0 through <math>p-1</math> exceeds the <b>docsDistributionMax</b> value for index partition <math>p</math>.</p> <p>An example of this would be a system with the following configuration:</p> <ul style="list-style-type: none"> <li>▪ Five index partitions.</li> <li>▪ A <b>docsDistributionMax</b> setting of "12500000,12500000,12500000,12500000,12500000".</li> <li>▪ A <b>triggers</b> setting of "10000,100000,1000000".</li> </ul> <p>The preceding configuration equates to index partition 1 re-indexing when index partition 0 reaches at least 10000 items, index partition 2 re-indexing when index partition 1 reaches at least 100,000 items, and index partition 3 re-indexing when index partition 2 reaches 1,000,000 items. Index partition 4 will be scheduled for re-indexing when the total number of items in index partitions 0 through 3 exceeds 12,500,000 items.</p>

## 2.2.4 CT\_post-processors

Referenced by: CT\_post-processors.

The **CT\_post-processors** complex type is used to call internal and external custom post-processors acting on the inverted index structures, as follows.

```
<xs:complexType name="CT_post-processors">
  <xs:sequence>
    <xs:element name="processor" type="CT_processor"/>
  </xs:sequence>
</xs:complexType>
```

### Child elements:

**processor:** The post-processor to call.

**Attributes:** None.

## 2.2.5 CT\_processor

Referenced by: <CT\_post-processors>

The **CT\_processor** complex type specifies internal and external post-processors to call after completion of an inverted index structure, as follows.

```
<xs:complexType name="CT_processor">
  <xs:attribute name="simple" type="xs:boolean"/>
  <xs:attribute name="external" type="xs:string"/>
</xs:complexType>
```

**Child elements:** None.

### Attributes

Name	Description
<b>Simple</b>	A Boolean value that <b>MUST</b> be <b>true</b> to enable the <b>simple</b> post-processor, or <b>false</b> to disable it. The <b>simple</b> post-processor verifies the file sizes of the newly built index structures.
<b>External</b>	A string value specifying the absolute path of an external process that will be executed every time an index partition has been re-indexed. The process will be launched with the absolute path of the directory containing the newly built index structures as the first and only argument.

## 2.2.6 CT\_ft

Referenced by: CT\_ft.

The **CT\_ft** complex type specifies the configuration parameters related to fault-tolerance, as specified in [\[MS-FSIFT\]](#), as follows.

```
<xs:complexType name="CT_ft">
  <xs:attribute name="enabled" type="xs:boolean" use="required"/>
  <xs:attribute name="storagePath" type="xs:string" use="required"/>
```

```

    <xs:attribute name="idleHeartbeatTime" type="xs:int" use="required"/>
    <xs:attribute name="maxStorageSizeMB" type="xs:int" use="required"/>
  </xs:complexType>

```

**Child elements:** None.

#### Attributes

Name	Description
<b>Enabled</b>	A Boolean value that MUST be <b>true</b> if fault-tolerance is to be enabled, or <b>false</b> if it is not.
<b>storagePath</b>	A string value specifying the directory in which the fault-tolerance storage is placed.
<b>idleHeartbeatTime</b>	An integer value specifying how often the indexing node is to reassess its column role.
<b>maxStorageSizeMB</b>	An integer value specifying the maximum size, in megabytes, allowed for the fault-tolerance storage. If the value is exceeded, the fault-tolerance storage will be truncated, starting with the oldest content, until the size limit is met.

### 2.2.7 CT\_real-time-properties

*Referenced by:* CT\_real-time-properties.

The **CT\_real-time-properties** complex type is not in use and MUST be ignored, as follows.

```

<xs:complexType name="CT_real-time-properties">
</xs:complexType>

```

**Child elements:** None.

**Attributes:** None.

### 3 Structure Examples

The following example shows the default configuration for an enterprise search service. Directory paths are dependent on user choices made during system installation. The default configuration is devised to suit the vast majority of installations.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <options
    xmlEncoding          = ""
    docsDistributionMax  = "12500000,12500000,12500000,12500000,12500000"
    numberDocsPerFixml  = "200"
    docsDistributionSteps = "100"
    docsDistributionSamples = "100"
    fsearchCachePstDist = "5,15,5,15,4,15,6,0,4,6,11,0,14"
    maxMBCacheSize      = "1000"
    maxSetDirs          = "400000"
    maxFixmlFiles       = "50000000"
    fixmlPrefetchThreads = "20"
    diskSpaceMBWarning  = "3000"
    indexerSwapDetect   = "-k 600 -K 1200"
    debugLog            = "false"
    numberPartitions    = "5"
    exclusionlistInterval = "60"
    indexingThreads     = "4"
    maxActiveIndexingJobs = "3"
    removeCollectionBatchSize = "5000"
    normalizeMode       = "periodic"
    normalizeInterval   = "1800"
    fixmlPath           = "c:/fastsearch/data/data_fixml"
    indexDir            = "c:/fastsearch/data/data_index"
    maxQueueSize        = "262144000"
    maxMemPerDocIndex   = "500"
  />

  <network
    qualifiedHostName    = "www.examplehost.com"
    useStrictBind        = "false"
    basePort             = "13050"
  />

  <index-scheduling type          = "docCountArchive"
                    triggers      = "10000,100000,1000000"
  />

  <post-processors>
    <processor simple          = "true"/>
  </post-processors>

  <ft
    enabled              = "false"
    storagePath          = "c:/fastsearch/data/ftStorage"
    idleHeartbeatTime    = "120"
    maxStorageSizeMB     = "51200"
  />
  <real-time-properties/>
</configuration>
```

## 4 Security Considerations

None.



## 5 Appendix A: Full XML Schema

For ease of implementation, the following full W3C XML schema for the elements, complex types, and attributes specified in the preceding sections is provided. Any schema references to namespaces included in ISO/IEC-29500:2008 refer specifically to the transitional schemas, as described in [\[ISO/IEC-29500-4\]](#).

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- ***** Global elements ***** -->

  <xs:element name="configuration" type="CT_configuration" />

  <!-- ***** Complex types ***** -->

  <xs:complexType name="CT_configuration">
    <xs:sequence>
      <xs:element name="options" type="CT_options"/>
      <xs:element name="network" type="CT_network"/>
      <xs:element name="index-scheduling" type="CT_index-scheduling"/>
      <xs:element name="post-processors" type="CT_post-processors"/>
      <xs:element name="ft" type="CT_ft"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_options">
    <xs:attribute name="xmlEncoding" type="xs:string" use="required"/>
    <xs:attribute name="docsDistributionMax" type="xs:string" use="required"/>
    <xs:attribute name="numberDocsPerFixml" type="xs:int" use="required"/>
    <xs:attribute name="docsDistributionSteps" type="xs:int" use="required"/>
    <xs:attribute name="docsDistributionSamples" type="xs:int" use="required"/>
    <xs:attribute name="fsearchCachePstDist" type="xs:string" use="required"/>
    <xs:attribute name="maxMBCacheSize" type="xs:string" use="required"/>
    <xs:attribute name="maxSetDirs" type="xs:int" use="required"/>
    <xs:attribute name="maxFixmlFiles" type="xs:int" use="required"/>
    <xs:attribute name="fixmlPrefetchThreads" type="xs:int" use="required"/>
    <xs:attribute name="diskspaceMBWarning" type="xs:int" use="required"/>
    <xs:attribute name="indexerSwapDetect" type="xs:string" use="required"/>
    <xs:attribute name="debugLog" type="xs:boolean" use="required"/>
    <xs:attribute name="numberPartitions" type="xs:int" use="required"/>
    <xs:attribute name="exclusionlistInterval" type="xs:int" use="required"/>
    <xs:attribute name="indexingThreads" type="xs:int" use="required"/>
    <xs:attribute name="maxActiveIndexingJobs" type="xs:int" use="required"/>
    <xs:attribute name="removeCollectionBatchSize" type="xs:int" use="required"/>
    <xs:attribute name="normalizeMode" type="xs:string" use="required"/>
    <xs:attribute name="normalizeInterval" type="xs:int" use="required"/>
    <xs:attribute name="fixmlPath" type="xs:string" use="required"/>
    <xs:attribute name="indexDir" type="xs:string" use="required"/>
    <xs:attribute name="maxQueueSize" type="xs:int" use="required"/>
    <xs:attribute name="maxMemPerDocIndex" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_network">
    <xs:attribute name="qualifiedHostName" type="xs:string" use="required"/>
    <xs:attribute name="useStrictBind" type="xs:string" use="required"/>
    <xs:attribute name="basePort" type="xs:string" use="required"/>
  </xs:complexType>

```

```
</xs:complexType>

<xs:complexType name="CT_index-scheduling">
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="triggers" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="CT_post-processors">
  <xs:sequence>
    <xs:element name="processor" type="CT_processor"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_processor">
  <xs:attribute name="simple" type="xs:boolean"/>
  <xs:attribute name="external" type="xs:string"/>
</xs:complexType>

<xs:complexType name="CT_ft">
  <xs:attribute name="enabled" type="xs:boolean" use="required"/>
  <xs:attribute name="storagePath" type="xs:string" use="required"/>
  <xs:attribute name="idleHeartbeatTime" type="xs:int" use="required"/>
  <xs:attribute name="maxStorageSizeMB" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="CT_real-time-properties">
</xs:complexType>

</xs:schema>
```

## 6 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® FAST™ Search Server 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

## 7 Change tracking

No table of changes is available. The document is either new or has had no changes since its last release.

## 8 Index

### A

[Applicability](#) 8

### C

[Change tracking](#) 20

Common data types and fields ([section 2 9](#), [section 2 9](#))

Complex types

[CT\\_ft](#) ([section 2.2.6](#) 13, [section 2.2.7](#) 14)

[CT\\_index-scheduling](#) 12

[CT\\_network](#) 11

[CT\\_options](#) 9

[CT\\_post-processors](#) 13

[CT\\_processor](#) 13

[CT\\_ft complex type](#) ([section 2.2.6](#) 13, [section 2.2.7](#) 14)

[CT\\_index-scheduling complex type](#) 12

[CT\\_network complex type](#) 11

[CT\\_options complex type](#) 9

[CT\\_post-processors complex type](#) 13

[CT\\_processor complex type](#) 13

### D

Data types and fields - common ([section 2 9](#), [section 2 9](#))

Details

common data types and fields ([section 2 9](#), [section 2 9](#))

[CT\\_ft complex type](#) ([section 2.2.6](#) 13, [section 2.2.7](#) 14)

[CT\\_index-scheduling complex type](#) 12

[CT\\_network complex type](#) 11

[CT\\_options complex type](#) 9

[CT\\_post-processors complex type](#) 13

[CT\\_processor complex type](#) 13

[global elements structure](#) 9

### E

[Example](#) 15

[Examples](#) 15

### F

[Fields - vendor-extensible](#) 8

Full XML schema ([section 5 17](#), [section 5 17](#))

### G

[Global elements structure](#) 9

[Glossary](#) 4

### I

[Implementer - security considerations](#) 16

[Informative references](#) 5

[Introduction](#) 4

### L

[Localization](#) 8

### N

[Normative references](#) 4

### O

[Overview \(synopsis\)](#) 5

[fault-tolerance](#) 7

[index scheduling](#) 7

[network](#) 7

[options](#) 6

[post processors](#) 7

### P

[Product behavior](#) 19

### R

[References](#) 4

[informative](#) 5

[normative](#) 4

[Relationship to protocols and other structures](#) 8

### S

[Security - implementer considerations](#) 16

Structure overview

[fault-tolerance](#) 7

[index scheduling](#) 7

[network](#) 7

[options](#) 6

[post processors](#) 7

Structures

[global elements](#) 9

overview ([section 2 9](#), [section 2 9](#))

### T

[Tracking changes](#) 20

### V

[Vendor-extensible fields](#) 8

[Versioning](#) 8

### X

XML schema ([section 5 17](#), [section 5 17](#))