[MS-FSCCFG]:
Crawler Configuration File Format

Release: July 16, 2012

Revision Summary

Date	Revision History	Revision Class	Comments
11/06/2009	0.1	Major	Initial Availability
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited 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

1 Introduction

This document specifies the Crawler Configuration File Format, an XML-based configuration format for a Web crawler process. This file format specifies configuration parameters that control the gathering, processing, and storage of information automatically retrieved by the Web crawler process from web sites, and then transmitting it to a search engine index.

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]:

checksum
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Internet Protocol version 4 (IPv4)
Internet Protocol version 6 (IPv6)
IPv4 address in string format
IPv6 address in string format
NT LAN Manager (NTLM) Authentication Protocol
path
realm
Secure Sockets Layer (SSL)
UTF-8
XML

The following terms are defined in [MS-OFCGLOS]:

Advanced Encryption Standard (AES)
cookie
crawl collection
crawl queue
crawl refresh cycle
crawl routing
document
duplicate server
file
File Transfer Protocol (FTP)
focused crawl
forms authentication
host name
HTTP POST
Hypertext Markup Language (HTML)
Hypertext Transfer Protocol 1.1 (HTTP/1.1)
MIME type
multinode scheduler
node scheduler
RSS channel
start URI
Uniform Resource Identifier (URI)
user name
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. 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.

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, http://www.w3.org/TR/html4/

[ISO-639-1] International Organization for Standardization, "Codes for the representation of names of languages -- Part 1: Alpha-2 code", 2002, http://www.iso.org/iso/catalogue_detail?csnumber=22109

[MC-RegEx] Microsoft Corporation, "Regular Expression Language Elements", http://msdn.microsoft.com/en-us/library/az24scfc(VS.80).aspx

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-FSID] Microsoft Corporation, "Indexing Distribution Protocol Specification".

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication 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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., et al., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, http://www.ietf.org/rfc/rfc2617.txt

[RFC3602] Frankel, S., Glenn, R., and Kelly, S., "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003, http://www.ietf.org/rfc/rfc3602.txt

[RFC959] Postel, J., and Reynolds, J., "File Transfer Protocol (FTP)", RFC 959, October 1985, http://www.ietf.org/rfc/rfc959.txt

[ROBOTSTXT] Koster, M., "A Method for Web Robots Control", November 1996, http://www.robotstxt.org/norobots-rfc.txt

[SITEMAPS] Sitemaps Org, "Sitemaps XML format", http://sitemaps.org/protocol.php

[SSL3] Netscape, "SSL 3.0 Specification", http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00

If you have any trouble finding [SSL3], please check here.

[WML2.0] Wireless Application Protocol Forum, Ltd., "Wireless Markup Language Version 2.0", Version 11-Sep-2001, http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

1.2.2 Informative References

[MS-FSCADM] Microsoft Corporation, "Crawler Administration and Status Protocol Specification".

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

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

1.3 Structure Overview (Synopsis)

This structure specifies the configuration **file** format and syntax used by a **Web crawler**. This XML-based format is a set of configuration parameters that control the behavior of a **crawl collection**. Collections are created or updated using these parameters.

1.4 Relationship to Protocols and Other Structures

The file format specified in this **document** is used by the protocol described in the Crawler Administration and Status Protocol Specification [MS-FSCADM].

1.5 Applicability Statement

None.

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Structures

The Web crawler process is configured and updated using an **XML**-based configuration file. The XML version MUST be 1.0 and it MUST be encoded in **UTF-8**.

2.1 Global Elements

The following are specified global elements.

2.1.1 CrawlerConfig

This element specifies that the XML following it is a Web crawler configuration object, as follows.

```
<xs:element name="CrawlerConfig" type="CT_CrawlerConfig" />
```

A Web crawler configuration file MUST contain one and only one CrawlerConfig XML element.

2.2 Complex Types

The following are complex type specifications.

2.2.1 CT_CrawlerConfig

This complex type referenced by **CrawlerConfig** specifies a crawl collection, as follows.

```
<xs:complexType name="CT_CrawlerConfig >
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="DomainSpecification" type="CT_DomainSpecification"/>
    </xs:choice>
</xs:complexType>
```

DomainSpecification: A **CT_attrib** element specifying a crawl collection.

2.2.2 CT_DomainSpecification

This complex type referenced by **CT_CrawlerConfig** specifies a crawl collection, as follows.

```
<xs:complexType name="CT_DomainSpecification">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="attrib" type="CT_attrib" maxOccurs="unbounded"/>
    <xs:element name="section" type="CT_section"/>
    <xs:element name="SubDomain" type="CT_SubDomain"/>
    <xs:element name="Login" type="CT_Login"/>
    <xs:element name="Node" type="CT_Node"/>
  </xs:choice>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
```

attrib: A **CT_attrib** element specifying a configuration parameter.

section: A **CT_section** element specifying a group of configuration parameters.

SubDomain: A **CT_SubDomain** element specifying a subsection.

Login: A **CT_Login** element specifying a login for **HTML forms authentication**.

Node: A **CT_Node** element.

The attribute of **CT_DomainSpecification** is found in the following table.

XML attribute	Type	Meaning
name	**xs:string**	The name of a crawl collection.

2.2.3 CT_attrib

This complex type referenced by **CT_DomainSpecification** and **CT_section** specifies a configuration parameter, as follows.

```
<xs:complexType name="CT_attrib" mixed="true">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="member" type="ST_member"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="ST_type" type="xs:string" use="required"/>
</xs:complexType>
```

member: An **ST_member** element specifying a string. This element MUST only be used if the **type** attribute is set to **list-string**, as specified in section 2.3.2.

Attributes of **CT_attrib** are found in the following table.

Attribute	Type	Meaning
name	**xs:string**	This specifies the name of the configuration parameter.
ST_type	**xs:string**	This specifies the data type of the configuration parameter. The **type** attribute MUST be one of the data types defined in the section element.

The following table specifies XML attribute values for the **attrib** element, which is of **CT_attrib** type, **as specified** in **CT_DomainSpecification** section 2.2.2**.** The first two columns in the following table specify the **name** and **type** XML attributes of an **attrib** XML element. The third column contains the default value of the configuration parameter. If the **type** XML attribute has a data type of **list-string**, as specified in section 2.3.2, then its default values are represented by using comma separated value format. The value *N/A* specifies that the configuration parameter does not have a default value. The last column specifies the purpose of the configuration parameter, whether it is associated with other configuration parameters, and describes its values.

Name	Type	Default	Meaning
info	string	N/A	Specifies meta information about this crawl collection.
fetch_timeout	integer	300	Specifies the maximum downloading time, in seconds, for a web document.

Name	Type	Default	Meaning
allowed_types	list-string	text/html, text/plain, application/ msword, application/ msexcel, application/ppt, application/pdf	Specifies valid web document **MIME types**. The Web crawler process discards other MIME types. This configuration parameter supports wildcard expansion of an entire field. Wildcards are represented by an asterisk character, for example, "text/*" or "*/*".
force_mimetype_detection	**boolean**	no	Specifies that the Web crawler process uses its own MIME type detection on documents.
allowed_schemes	list-string	http	Specifies the **URI** schemes, as specified in [RFC2396], that the web crawler MUST process.
ftp_passive	**boolean**	yes	Specifies that the Web crawler uses passive **FTP** mode, as specified in [RFC959].
domain_clustering	**boolean**	no	Specifies the Web crawler to route hosts from the same domain to one node scheduler in a multi node installation.
max_inter_docs	integer	N/A	Specifies the maximum number of documents that can be crawled on one crawl site prior to processing a new crawl site. If this limit is reached a new crawl site will be crawled, thus interleaving crawling of the crawl sites.
max_redirects	integer	10	Specifies the maximum number of **HTTP** redirects to follow from a URI.
diffcheck	**boolean**	yes	Specifies that the Web crawler performs duplicate detection. The duplicate detection is performed by checking whether two or more web documents have the same content.
near_duplicate_detection	**boolean**	no	Specifies that the Web crawler MUST use a less strict duplicate detection algorithm. In this case duplicate documents are detected by identifying a unique pattern of words.
max_uri_recursion	integer	5	Specifies the maximum number of times a pattern can be appended to successors of a URI.
ftp_searchlinks	**boolean**	yes	Specifies that the Web crawler MUST search for hyperlinks in documents downloaded from FTP servers.
use_javascript	boolean	no	Specifies that the Web crawler MUST process JavaScript that is contained in HTML documents.

Name	Type	Default	Meaning
javascript_keep_html	boolean	no	Specifies what to submit to the indexing engine. If this parameter is set to *yes*, the HTML resulting from the JavaScript processing is used. Otherwise, the original HTML document is used. This option MUST NOT be used if the **use_javascript** configuration parameter is not set to *yes*.
javascript_delay	real	N/A	Specifies the delay, in seconds, to use when retrieving dependencies associated with an HTML document with JavaScript. If this configuration parameter is not specified, the Web crawler MUST use the **delay** configuration parameter to fetch external JavaScript documents.
exclude_exts	list-string	.jpg, .jpeg, .ico, .tif, .png, .bmp, .gif, .wmf, .avi, .mpg, .wmv, .wma, .ram, .asx, .asf, .mp3, .wav, .ogg, .ra, .aac, .m4a, .zip, .gz, .vmarc, .z, .tar, .iso, .img, .rpm, .cab, .rar, .ace, .hqx, .swf, .exe, .java, .jar, .prz, .wrl, .midr, .css, .ps, .ttf, .mso, .dvi	Specifies file extensions that MUST be excluded by the crawl.
use_http_1_1	boolean	yes	Specifies that the Web crawler MUST use **HTTP/1.1**.
accept_compression	boolean	yes	Specifies that the Web crawler MUST accept compression. This parameter has no effect if the **use_http_1_1** configuration parameter is not enabled.
dbswitch	integer	5	Specifies the number of crawl refresh cycles a web document can have without being processed by the Web crawler. The **dbswitch_delete** configuration parameter specifies the action to perform on expired documents.
dbswitch_delete	boolean	no	Specifies that the Web crawler MUST delete URIs that were not crawled in **dbswitch**

Name	Type	Default	Meaning
			crawl refresh cycles.
html_redir_is_redir	boolean	yes	Specifies that the Web crawler MUST treat documents as HTTP redirects if they are associated with a refresh HTML **META/** tag as specified in [HTML].
hmtl_redir_threshold	integer	3	Specifies the maximum number of seconds that a web document with an HTML **META/** tag can be treated as an HTTP redirect. This configuration parameter MUST be ignored if the **html_redir_is_redir** configuration parameter is not set.
robots_ttl	integer	86400	Specifies how frequently the Web crawler MUST retrieve the robots.txt file, as specified in [ROBOTSTXT], from a crawl site. This frequency must be specified in seconds.
use_sitemaps	boolean	no	Specifies whether the Web crawler MUST use sitemaps, as specified in [SITEMAPS]. If this parameter is enabled, the Web crawler finds and parses sitemaps. The sitemap information is used if the **refresh_mode** configuration parameter is set to "adaptive", in which case the Web crawler uses the change frequency information, otherwise it only extracts and follows links from sitemaps.
max_pending	integer	2	Specifies the maximum number of outstanding HTTP requests for a crawl site.
robots_auth_ignore	boolean	yes	Specifies whether or not the Web crawler MUST ignore robots.txt, as specified in [ROBOTSTXT], if a 401/403 HTTP authentication error is returned by the server. If disabled, the crawler will not crawl the site.
robots_tout_ignore	boolean	no	Specifies whether the Web crawler MUST ignore rules from the robots.txt file if the request for this file times out.
rewrite_rules	list-string	N/A	Specifies a set of rules that are used to rewrite URIs. A rewrite rule has two components: an expression to match (match_pattern), and a replacement string (replacement_string) that MUST replace the first expression. The expression to match is a grouped match regular expression, as specified in [MC-RegEx]. The format of the rewrite rule is "*match_pattern*replacement_string*", where * is any character that is not white space.
extract_links_from_dupes	boolean	no	Specifies that the Web crawler MUST extract hyperlinks from documents.

Name	Type	Default	Meaning
use_meta_csum	boolean	no	Specifies that the Web crawler uses META tags, as specified in [HTML], to generate a duplicate detection **checksum**.
csum_cut_off	integer	0	Specifies the maximum number of bytes to use to generate the duplicate detection checksum. If this parameter is set to 0, the feature is disabled.
if_modified_since	boolean	yes	Specifies whether the Web crawler MUST send HTTP headers that contain a value of "If-Modified-Since".
use_cookies	boolean	no	Specifies whether the Web crawler MUST send and store **cookies**.
uri_search_mime	list-string	text/html, text/ vnd.wap.xml, text/wml, text/x-wap.wml, x-application/ wml, text/x-hdml	Specifies the MIME types from which the Web crawler extracts hyperlinks. This configuration parameter supports wildcard expansion only at entire field level. A wildcard is represented by the asterisk character, for example, "text/*" or "*/*".
max_backoff_counter	integer	50	Specifies the maximum number of crawl site connection failures. If the number of failed connections exceeds this limit, then the crawling process will stop on this crawl site.
max_backoff_delay	integer	600	Specifies the maximum delay, in seconds, for a crawl site when network problems occur. When sites have network failures, the Web crawler increases the fetch delay to no more than this amount.
delay	real	60.0	Specifies how frequently the Web crawler can retrieve a document from a crawl site. This parameter is represented in seconds.
refresh	real	1500.0	Specifies how frequently the Web crawler MUST perform a crawl refresh cycle. This parameter is represented in minutes.
robots	boolean	yes	Specifies that the Web crawler MUST obey the rules found in robot.txt files, as specified in [ROBOTSTXT].
start_uris	list-string	N/A	Specifies **start URI**s for the Web crawler.
start_uri_files	list-string	N/A	Specifies a list of files that contain start URIs. These files are stored in plain text file format, with one start URI per line.
max_sites	integer	128	Specifies the maximum number of sites to

Name	Type	Default	Meaning
			crawl concurrently.
mirror_site_files	list-string	N/A	Specifies a list of files that contain mirror sites for a specified domain. A mirror site is a replica of an already existing crawl site. This file MUST use following format: a plain text file with a space-separated list of crawl sites, with the preferred name listed first.
proxy	list-string	N/A	Specifies a set of proxies that the Web crawler MUST use to fetch documents. Each proxy is specified using the following format: "(http://)(username:password@)hostname(:port)", optional parts are contained within parentheses. The password is encrypted as specified in section 2.2.4.20.
proxy_max_pending	integer	Maximum value of INT32, as specified in [MS-DTYP].	Specifies a limit on the number of outstanding open connections per proxy.
headers	list-string	User-Agent: FAST Search Web Crawler <*version*>	Specifies additional HTTP headers to add to the request sent to the web servers.
cut_off	integer	N/A	Specifies the maximum number of bytes in a document. A web document larger than this size limit is discarded or truncated depending on the value of the **truncate** configuration parameter. If no **cut_off** configuration parameter is specified, this option is disabled.
truncate	boolean	yes	Specifies whether a web document MUST be truncated when a web document exceeds the specified **cut_off** threshold.
check_meta_robots	boolean	yes	Specifies that the Web crawler MUST follow the directives given by the **NoIndex /** and **NoFollow /** META tags as specified in [HTML].
obey_robots_delay	boolean	no	Specifies that the Web crawler MUST follow the crawl-delay directive in robots.txt files, as specified in [ROBOTSTXT].
key_file	string	N/A	Specifies the **path** to an **SSL** key file used for **HTTPS** connections as specified in [SSL3].
cert_file	string	N/A	Specifies the path to a X509 certificate file used for HTTPS connections as specified in [X509].

Name	Type	Default	Meaning
max_doc	integer	100000	Specifies the maximum number of documents to download from a site.
enforce_delay_per_ip	boolean	yes	Specifies that the Web crawler limits requests to web servers whose names map to a shared **IPv4** or **IPv6** address. This parameter depends on the **delay** configuration parameter.
wqfilter	boolean	yes	Specifies whether the Web crawler MUST use a filter that removes duplicate URIs from the **crawl queue**s.
smfilter	integer	0	Specifies the maximum number of bits in the bloom filter that removes duplicate URIs from the queue associated with the node scheduler. A bloom filter is a space-efficient probabilistic data structure, a bit array, which is used to test if an element is a member of a given set. The test may yield a false positive but never a false negative
mufilter	integer	0	Specifies the maximum number of bits used in the bloom filter, see **smfilter,** which removes duplicate URIs, which are sent from a **node scheduler** to a **multinode scheduler**.
umlogs	boolean	yes	Specifies whether all logging is sent to the multinode scheduler for storage. If this parameter is not enabled, logs reside only on the node schedulers.
sort_query_params	boolean	no	Specifies whether the Web crawler MUST sort the parameters in the query component of a URI, as specified in [RFC2396]. Typically, query components are key-value pairs that are separated by semicolons or ampersands. When this configuration parameter is set, the query is sorted alphabetical by the key name.
robots_timeout	integer	300	Specifies the maximum number of seconds that the Web crawler can use to download a robots.txt file.
login_timeout	integer	300	Specifies the maximum number of seconds that the Web crawler can use for a login request.
send_links_to	string	N/A	Specifies a crawl collection name to which all extracted hyperlinks is sent.
cookie_timeout	integer	900	Specifies the maximum number of seconds a session cookie is stored. A session cookie is a cookie with no expiration date.
refresh_when_idle	boolean	no	Specifies whether the Web crawler MUST trigger a new crawl refresh cycle when it

Name	Type	Default	Meaning
			becomes idle. This option MUST NOT be used in a multinode installation.
refresh_mode	string	scratch	Specifies the refresh mode of a crawl collection. There are five valid values for the **refresh_mode** configuration parameter: **append:** Add the start URIs to the end of the crawl queue when a crawl refresh cycle begins. **prepend:** Add the start URIs to the beginning of the crawl queue when a crawl refresh cycle begins. **scratch:** Truncate the crawl queue previous to appending the start URIs to the queue. **soft:** If the crawl queue for a site is not empty at the end of a crawl refresh cycle, the Web crawler continues crawling into the next crawl refresh cycle. A site is not refreshed until the crawl queue is empty. **adaptive:** Build crawl queue according to the adaptive section configuration parameters specified in section 2.2.4.24.

2.2.4 CT_section

This complex type referenced by **CT_DomainSpecification** aggregates a set of **CT_attrib** XML elements, as specified in section 2.2.3 that are logically associated with each other. Sections can be nested, as follows.

```
<xs:complexType name="CT_section">
   <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attrib" type="CT_attrib"/>
      <xs:element name="section" type="CT_section"/>
  </xs:choice>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
```

attrib: A **CT_attrib** element specifying a configuration parameter.

section: A **CT_section** element specifying a set of configuration parameters and, or sections.

The attributes of a **CT_section** MUST contain the XML attribute specified in the following table.

XML attribute	Type	Meaning
name	xs:string	A string specifying an unique name of the logically grouped **attrib** XML elements.

The following sections specify XML attribute values for the **section** element, which is of **CT_section** type, in **CT_DomainSpecification** as specified in section 2.2.2. For each section valid **attrib** and **section** values are specified.

2.2.4.1 include_domains Section

This section is a set of domain based rules that specify which URIs to include in a crawl collection. An empty section MUST match all domains. The following table specifies **attrib** elements for this section.

Name	Type	Default	Meaning
exact	list-string	N/A	Specifies a list of **host names**. If the domain name of a URI matches exactly one of these host names, the URI is included by this rule.
prefix	list-string	N/A	Specifies a list of host names. If the domain name of a URI begins with one of these host names, the URI is included by this rule.
suffix	list-string	N/A	Specifies a list of host names. If the domain name of a URI ends with one of these host names, the URI is included by this rule.
regexp	list-string	N/A	Specifies a list of regular expressions [MC-RegEx]. If the domain name matches one of these regular expressions, the URI is included by this rule.
ipmask	list-string	N/A	Specifies a list of IPv4 address masks. If the IPv4 address of a URI that was retrieved matches one of these IPv4 address masks, the URI is include by this rule. An IPv4 address mask MUST follow one of the following formats: A range of IPv4 address can be specified by writing an **IPv4 address in string format** and using a hyphen for the range, for example, 207.46.197.0-100 or 207.46.190-197.100. If an IPv4 address is within this range, it is included by this mask. A IPv4 mask can also be specified by examining the *N* most significant bits of an IPv4 address, where *N* is within the range of {0, 32}. The mask is an IPv4 address in string format followed by a forward slash and the number of most significant bits, for example, 207.46.197.0 /24. If an IPv4 address has the same *N* bits of the specified IPv4 address, it is included by this mask. A IPv4 mask can also be specified by using a bit mask to mask out important bits of an IPv4 address. The format of this mask is "IPv4 address in string format:ip-mask", where ip-mask is an IPv4 address in string format used for masking or a 32 bit hexadecimal digit, for example, 207.46.197.0:255.255.255.0 or 207.46.197.0:0xffffff00. If an IPv4 address has the same bits set as specified by the ip-mask and the IPv4 address, it is included by this mask.
ip6mask	list-string	N/A	Specifies a list of IPv6 address masks. If the IPv6 address of a URI that was retrieved matches one of these IPv6 address masks, the URI is included by this rule. An IPv6 address mask MUST follow one of the following formats: A range of IPv6 address can be specified by writing an **IPv6 address in string format** and using a hyphen for the range, for example 2002:CF2E:C500- C564:0:0:0:0:0 or ::ffff:207.46.197.0-100. If an IPv6 address is within this range it is included by this mask. A IPv6 mask can also be specified by looking at the *N* most significant bits of an IPv6 address, where *N* has the range of {0, 128}. This mask is an IPv6 address in string format followed by a forward slash and the

Name	Type	Default	Meaning
			number of most significant bits, for example, 2002:CF2E:C500:0:0:0:0:0/60. If an IPv6 address has the same *N* bits of the specified IPv6 address, it is included by this mask.
file	list-string	N/A	Specifies a list of files that contains a set of rules. These files are stored in plain text file format, with one rule per line. Each rule is given on the following format "ruletype:rule". Valid ruletype values are: "exact", "prefix", "suffix", "regexp", "ipmask" or "ip6mask".

2.2.4.2 exclude_domains Section

This section is a set of domain based rules that specify which URIs to exclude from a crawl collection. An empty section MUST NOT match any domains. The table in section 2.2.4.1 specifies the **attrib** elements for this section.

2.2.4.3 include_uris Section

This section is a set of URI based rules that specify which URIs to include in a crawl collection. An empty section MUST match all URIs. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
exact	list-string	N/A	Specifies a list of URIs. If a URI exactly matches one of these URIs, the URI is included by this rule.
prefix	list-string	N/A	Specifies a list of strings. If a URI begins with one of these strings, the URI is included by this rule.
suffix	list-string	N/A	Specifies a list of strings. If a URI ends with one of these strings, the URI is included by this rule.
regexp	list-string	N/A	Specifies a list of regular expressions [MC-RegEx]. If a URI matches one of these regular expressions, the URI is included by this rule.
file	list-string	N/A	Specifies a list of files that contains a set of rules. These files are stored in plain text file format, with one rule per line. Each rule is given on the following format "ruletype:rule". Valid ruletype values are: "exact", "prefix", "suffix" or "regexp".

2.2.4.4 exclude_uris Section

This section is a set of URI based rules that specify which URIs to exclude from a crawl collection. An empty section MUST NOT match any URIs. The table in section 2.2.4.3 specifies the **attrib** elements for this section.

2.2.4.5 log Section

This section specifies logging behavior for the Web crawler process.

Possible values are described in the following table.

Name	Type	Default	Meaning
fetch	string	text	Enable or disable logging of downloaded documents. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **none:** This disables logging.
postprocess	string	text	Enable or disable logging of node scheduler post processing of the documents. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **xml:** This creates an XML formatted log. ▪ **none:** This disables logging.
header	string	none	Enable or disable logging of HTTP headers. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **none:** This disables logging.
screened	string	none	Enable or disable logging of all screened URIs. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **none:** This disables logging.
scheduler	string	none	Enable or disable logging of adaptive crawling. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **none:** This disables logging.
dsfeed	string	text	Enable or disable logging of feeding data to the indexing engine. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **none:** This disables logging.
site	string	text	Enable or disable logging per site. Valid values are: ▪ **text:** This creates a text formatted log. ▪ **none:** This disables logging.

2.2.4.6 storage Section

This section specifies how the Web crawler stores data and metadata. The following table specifies the **attrib** XML elements for this section.

Name	Type	Default	Meaning
datastore	string	bstore	Specifies the format for web document data storage. Valid values are: ▪ **flatfile:** This format stores documents directly into the file system. ▪ **bstore:** This format partition documents into fixed sized blocks and distributes them across a set of files. An index maps the order of the blocks, and specifies which blocks belong to a document.
store_http_header	boolean	yes	Specifies that the process MUST store the received HTTP header.
store_dupes	boolean	no	Specifies that the Web crawler MUST store duplicate documents.
compress	boolean	yes	Specifies that downloaded documents MUST be compressed previous to storing them.
compress_exclude_mime	list-string	none	Specifies a set of MIME types. Downloaded documents that match these MIME types MUST NOT be compressed. If the **compress** configuration parameter is not set, this parameter is not applicable.
remove_docs	boolean	no	Specifies that the Web crawler MUST delete documents that were submitted to the indexing engine.
clusters	integer	8	Specifies the number of clusters to use for storage in a crawl collection. Web documents are distributed among these storage clusters.
defrag_threshold	integer	85	Specifies defragmentation thresholds as a percentage. Data storage can become fragmented; defragmentation must be performed to reclaim fragmented space. The default value means that the reclaimable space in the data storage file is 100% minus the default value. If the value specified in this parameter is reached, the Web crawler triggers defragmentation of a particular file.
uri_dir	string	none	Specifies a path to store URIs that are extracted from a document. A file is generated for each crawled document.

2.2.4.7 pp Section

This section specifies the post processing behavior for a node scheduler. Post processing consists of two primary tasks; feeding web documents to the index and performing duplicate detection. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
use_dupservers	boolean	no	Specifies that the Web crawler MUST use one or more **duplicate servers**. This option is applicable only in a multinode installation.
max_dupes	integer	10	Specifies the maximum number of duplicates to record.
stripe	integer	1	Specifies the number of files to spread data to. This is done in order improve the performance of the post processing database.
ds_meta_info	list-string	duplicates, redirects, mirrors, metadata	Specifies the type of metadata a node scheduler MUST report to the indexing engine. Valid values are: ▪ **duplicates:** Report URIs that are duplicates of this document. ▪ **redirects:** Report URIs that are redirected to this document. ▪ **metadata:** Report meta data of this document. ▪ **mirrors:** Report all mirror URIs of this web document.
ds_max_ecl	integer	10	Specifies the maximum number of duplicates or redirects to report to the indexing engine, as specified by the **ds_meta_info** configuration parameter.
ecl_override	string	N/A	Specifies a regular expression [MC-RegEx]. URIs that match this expression, and are associated with a specific status, are submitted to the indexing engine whether or not the max_dupes configuration parameter limit has been reached. The URI MUST have a status of duplicates or redirects.
ds_send_links	boolean	no	Specifies whether all extracted hyperlinks from a web document MUST be sent to the indexing engine.
ds_paused	boolean	no	Specifies whether a node scheduler MUST pause content submission of data to the indexing engine.

2.2.4.8 ppdup Section

This section specifies duplicate server settings. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
format	string	N/A	Specifies the duplicate server database format. Valid values are: ▪ **gigabase:** A simple key-value database. ▪ **hashlog:** An in-memory data structure consisting of a hash table and a log. The log ensures the rebuilding of the hash table. ▪ **diskhashlog:** The same as **hashlog**, except that the data

Name	Type	Default	Meaning
			structure is persisted to disk.
cachesize	integer	N/A	Specifies the duplicate server database cache size in megabytes. If the format configuration parameter is set to **hashlog** or **diskhashlog** this parameter specifies the initial size of the hash table.
stripes	integer	N/A	Specifies the number of files to spread data to. This is done in order improve the performance of the duplicate server database.
compact	boolean	yes	Specifies whether or not the duplicate server database MUST perform compaction.

2.2.4.9 feeding Section

The **feeding** section MUST consist of at least one section element that specifies how to send a representation of the crawl collection to the indexing engine. Such a section defines a content destination. The **name** attribute specifies a unique name for the content destination. The following table specifies the **attrib** elements for a content destination section.

Name	Type	Default	Meaning
collection	string	N/A	Specifies the name of the content collection to which to submit documents. This configuration parameter MUST be specified within a feeding section.
destination	string	default	Reserved. This configuration parameter MUST contain the value "default".
paused	boolean	no	Specifies whether the web crawler suspend submission of content to the indexing engine.
primary	boolean	N/A	Specifies whether this content destination is a primary or secondary content destination. A primary content destination can act on callback information, as specified in [MS-FSID], during content submission to the indexing engine.

2.2.4.10 cachesize Section

The **cachesize** section MUST be used to configure the cache sizes for the Web crawler process. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
duplicates	integer	N/A	Specifies the maximum number of entries in the duplicate checksum cache.
screened	integer	N/A	Specifies the maximum number of entries in the screened URI cache.
smcomm	integer	N/A	Specifies the maximum number of entries in the cache used for communication within a node scheduler.
mucomm	integer	N/A	Specifies the maximum number of entries in the cache for communication between a multinode scheduler and the crawler

Name	Type	Default	Meaning
			single node schedulers.
wqcache	integer	N/A	Specifies the maximum number of entries in the crawl queue for a site.
crosslinks	integer	N/A	Specifies the maximum number of entries in the crosslink cache. The crosslink cache contains retrieved hyperlinks and referring hyperlinks.
routetab	integer	1048576	Specifies the crawl routing database cache size, in bytes.
pp	integer	1048576	Specifies the post process database cache size, in bytes.
pp_pending	integer	131072	Specifies the post process pending cache size, in bytes. The pending cache contains entries that were not sent to the duplicate servers.
aliases	integer	1048576	Specifies the aliases data mapping database cache size, in bytes. A site can be associated with one or more aliases or alternative names.

2.2.4.11 http_errors Section

The **http_errors** section specifies how to handle various HTTP/HTTPS error response codes and conditions.

The following table specifies the **attrib** elements for this section and the **ftp_errors** section, specified in section 2.2.4.13, because both of these sections have the same format. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column.

Name	Type	Default	Meaning
The **name** XML attribute specifies the HTTP/HTTPS/FTP response code number to handle. The character **X** can be used as a wildcard, for example **4XX**. Other valid values are: **net:** Used to handle network socket errors. **int:** Used to handle internal error in the Web crawler. **ttl:** Used to handle HTTP/HTTPS/FTP connection time outs.	string	See section 2.2.4.12 and section 2.2.4.14.	Specifies how the Web crawler handles HTTP/HTTPS/FTP and network errors. Valid options for handling individual response codes are: ▪ **KEEP:** Keep the web document unchanged ▪ **DELETE[:X]:** Delete the web document if the error condition occurs for *X* retires. Deletion happens immediately if no *X* value is specified. If "RETRY[:X]" is specified for either of these options, the Web crawler will re-download the web document no more than *X* number of times in the same crawl refresh cycle period previous to failing the attempt.

2.2.4.12 Default Values for the http_errors Section

The following table specifies the default values for the **http_errors** section.

Name	Value
4xx	DELETE:0
5xx	DELETE:10
int	KEEP:0
net	DELETE:3, RETRY:1
ttl	DELETE:3

2.2.4.13 ftp_errors Section

This section specifies how to handle various response codes and error conditions for FTP URIs. Section 2.2.4.11 specifies the **attrib** elements and default values for this section.

2.2.4.14 Default Values for the ftp_errors Section

The following table specifies the default values for the **ftp_errors** section.

Name	Value
4xx	DELETE:3
550	DELETE:0
5xx	DELETE:3
int	KEEP:0
net	DELETE:3, RETRY:1

2.2.4.15 workqueue_priority Section

This section specifies the priority levels for the crawl queues, and specifies the rules and modes used to insert URIs into and extract URIs from the queues. The following table specifies **attrib** elements for this section.

Name	Type	Default	Meaning
levels	integer	1	Specifies the number of priority levels used for the crawl queues.
default	integer	1	Specifies a default priority level that is assigned to URIs in a crawl queue.
start_uri_pri	integer	1	Specifies the priority level for start URIs, see the start_uris and the start_uri_files configuration parameters specified in section 2.2.3.
pop_scheme	string	default	Specifies the mode the Web crawler uses to extract URIs from the crawl queue. Valid values are: ▪ **rr:** This mode extracts URIs from the priority levels in round-robin order. ▪ **wrr:** This mode extracts URIs from the priority levels in a weighted round-robin order. The weights are based on their

Name	Type	Default	Meaning
			respective share setting per priority level, as specified in section 2.2.4.13. - **pri:** This mode extracts URIs from the priority levels in priority order by when entries still remain in the crawl queue. 1 is the highest priority, as specified in section 2.2.4.13. - **default:** This mode is the same as **wrr**.
put_scheme	string	default	Specifies which Web crawler mode to use when inserting URIs into the crawl queue. Valid values are: - **default:** This mode always inserts URIs with the priority level specified in the default configuration parameter. - **include:** This mode inserts URIs with the priority level of include_domains or include_uris, as specified in section 2.2.4.13 for every priority level. The Web crawler process assigns the default priority level when a URI does not match any of these sections.

2.2.4.16 Priority Level Sections

Within the **workqueue_priority** section, a set of sections that specify priority levels and weights of the crawler queues can be specified. Those sections will only be used if the **pop_scheme**, as specified in 2.2.4.15, is set to "wrr" or "pri". The **name** XML attribute of these sections MUST be the priority level to be specified. The priority levels MUST begin at 1.

The **include_domains** or **include_uris** sections can be used within each priority level section, as specified in sections 2.2.4.1 and 2.2.4.3 respectively. URIs that match these rules MUST be queued using the matching priority level. In addition, the following table specifies **attrib** elements for these sections.

Name	Type	Default	Meaning
share	integer	N/A	Specifies a weight to use for each crawl queue. This weight MUST only be used if the **pop_scheme** configuration parameter is set to "wrr".

2.2.4.17 link_extraction Section

The **link_extraction** section specifies which type of hyperlinks to follow.

All uppercase HTML **TAG/** references are specified in [HTML]. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
a	boolean	yes	Extract hyperlinks from **A/** HTML tags.
action	boolean	yes	Extract hyperlinks from action attributes in HTML tags.
area	boolean	yes	Extract hyperlinks from **AREA/** HTML tags.
card	boolean	yes	Extract hyperlinks from the **CARD/** Wireless Markup Language

Name	Type	Default	Meaning
			tags as specified in [WML2.0].
comment	boolean	yes	Extract hyperlinks from comments within a web document.
embed	boolean	yes	Extract hyperlinks from **EMBED/** HTML tags.
frame	boolean	yes	Extract hyperlinks from **FRAME/** HTML tags.
go	boolean	yes	Extract hyperlinks from **GO/** Wireless Markup Language tags as specified in [WML2.0].
img	boolean	no	Extract hyperlinks from **IMG/** HTML tags.
layer	boolean	yes	Extract hyperlinks from **LAYER/** HTML tags.
link	boolean	yes	Extract hyperlinks from **LINK/** HTML tags.
meta	boolean	yes	Extract hyperlinks from **META/** HTML tags.
meta_refresh	boolean	yes	Extract hyperlinks from **META http-equiv="refresh"/** HTML tags.
object	boolean	yes	Extract hyperlinks from **OBJECT/** HTML tags.
script	boolean	yes	Extract hyperlinks from **SCRIPT/** HTML tags.
script_java	boolean	yes	Extract hyperlinks from **SCRIPT/** HTML tags that contain JavaScript.
style	boolean	yes	Extract hyperlinks from **STYLE/** HTML tags.

2.2.4.18 limits Section

The **limits** section specifies fail-safe limits for a crawl collection. When the collection exceeds the limit it enters a crawl mode called *refresh only*. This mode specifies that only previously-crawled URIs are crawled again. The following table specifies **attrib** elements for this section.

Name	Type	Default	Meaning
disk_free	integer	0	Specifies the percentage of free disk space that must be available for the Web crawler to operate in normal crawl mode. If the percentage becomes less than this limit, the Web crawler enters the *refresh only* crawl mode. If the parameter is set to 0, this feature is disabled.
disk_free_slack	integer	3	Specifies, in percentage, an amount of disk space which can be used in addition to the disk space reserved by **disk_free**.
max_doc	integer	0	Specifies the number of stored documents required to trigger the *refresh only* crawl mode. If this parameter is set to 0 this feature is disabled.
max_doc_slack	integer	1000	Specifies the maximum number of documents that can be contained in a slack, up to the **max_doc** configuration parameter threshold.

2.2.4.19 focused Section

This section MUST be used to configure **focused crawl**. An **exclude_domains** section can be used within the **focused** section, as specified in section 2.2.4.1 to exclude domains from this focused scheduling. If no **exclude_domain** section is defined, all domains are included in the focused scheduling. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
languages	list-string	N/A	Specifies list of languages for documents that can be stored by the Web crawler, as specified in [ISO-639-1].
depth	integer	N/A	Specifies the number of page hops to follow for web documents that do not match the specified languages, as set by the languages configuration parameter.

2.2.4.20 passwd Section

This section configures credentials for sites that require authentication. The Web crawler supports basic authentication, as specified in [RFC2617], digest authentication, as specified in [RFC2617], and **NTLM** authentication.

The following table specifies the **attrib** XML elements for this section. Because there are multiple values for the **name** XML attribute, a description of each purpose is included in the name column.

Name	Type	Default	Meaning
The **name** XML attribute MUST contain a URI or **realm**. A valid URI behaves as a prefix value, because all hyperlinks extracted at its level or deeper use these authentication settings.	string	N/A	The credentials MUST be specified in one of the following formats: "username:password" or "usename:password:realm:scheme". The password component of the credential string can be encrypted; if not encrypted it is given in plain text. An encrypted password MUST begin with the string "!ENC!", and MUST be followed by the encrypted password. The password is encrypted using Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode, as specified in [RFC3602]. The encryption key file resides in "%FASTSEARCH%\etc\CrawlerEncryptionKey.dat". This file MUST contain a 128 bit key and a 128 bit initialization vector, as specified in [RFC3602]. If the credentials are given using the "username:password" format, the Web crawler automatically uses basic access authentication. Otherwise the configuration MUST specify an authentication scheme. Valid authentication schemes are: ▪ **basic:** Specifies that the Web crawler should use basic authentication. ▪ **digest:** Specifies that the Web crawler should use digest authentication. ▪ **ntlmv1:** Specifies that the Web crawler should use NTLMv1, as specified in [MS-NLMP]. ▪ **ntlmv2:** Specifies that the Web crawler should use NTLMv2, as specified in [MS-NLMP]. ▪ **auto:** Specifies that the Web crawler determines the

Name	Type	Default	Meaning
			authentication scheme by itself.

2.2.4.21 ftp_acct Section

This section specifies FTP accounts for crawling FTP URIs. The following table specifies the **attrib** elements for this section. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column.

Name	Type	Default	Meaning
The value of the **name** attribute is the site name for which this FTP account is valid.	string	N/A	This is the **user name** and password for this FTP account. The string MUST be of the format "username:password".

2.2.4.22 exclude_headers Section

The **exclude_headers** section MUST be used to specify documents to exclude from the crawl based on the contents of the HTTP header fields. The **attrib** elements for this section are specified in the following table. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column of the table.

Name	Type	Default	Meaning
The **name** XML attribute is used to set the name of the HTTP header to test.	list-string	N/A	Specifies a list of regular expressions [MC-RegEx]. If the value of the specified HTTP header matches one of these regular expressions, the web document is excluded from the crawl.

2.2.4.23 variable_delay Section

This section specifies time slots that use a different delay request rate. When no time slot is specified, the crawler uses the **delay** configuration parameter, as specified in section 2.2.3. The following table specifies the **attrib** elements for this section. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column.

Name	Type	Default	Meaning
The **name** attribute is used to set the time slot. The format of this string is DDD:HH.MM-DDD:HH.MM, for example "Mon:21.00-Mon:22.30".	string	N/A	Specifies the delay request rate for this time slot, in seconds. A value of "suspend" specifies that crawling of this crawl collection will be suspended.

2.2.4.24 adaptive Section

This section specifies the adaptive crawling options. The **refresh_mode** configuration parameter, specified in section 2.2.3, MUST be set to "adaptive" for this section to be used by the Web crawler. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
refresh_count	integer	4	Specifies the number of minor refresh cycles. A refresh cycle

Name	Type	Default	Meaning
			can be divided into a number of fixed size time intervals that are called minor refresh cycles.
refresh_quota	integer	90	Specifies the ratio of existing re-crawled URIs to new unseen URIs, expressed as a percentage. Setting the percentage low gives preference to new URIs.
coverage_min	integer	25	Specifies a minimum number of URIs to be crawled in a minor refresh cycle.
coverage_max_pct	integer	10	Specifies a limit percentile to site re-crawl within a minor refresh cycle.

The adaptive crawling behavior can be controlled with the sections that are specified in 2.2.4.25 and 2.2.4.26.

2.2.4.25 weights Section

In this section, each URI is given a score in the adaptive crawling process. This score is used to prioritize URIs and is based on a set of rules. Each rule is assigned a weight that determines its contribution towards the total score that is specified in the **weights** section. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
inverse_length	real	1.0	Specifies the weight for the inverse length rule. The inverse length rule gives URIs with less than 10 slashes a high score, and it gives no score to URIs with 10 or more slashes.
inverse_depth	real	1.0	Specifies the weight for the inverse depth rule. The number of page hops from a start URI is computed; a high score is assigned to URIs that have less than 10 page hops. The rule gives a score of zero for URIs with 10 or more page hops.
is_landing_page	real	1.0	Specifies the weight for the **is_landing_page** rule. This rule gives URIs that ends with a slash (/) or "index.html" a higher score. The rule gives no score to URIs that have query components.
is_mime_markup	real	1.0	Specifies the weight for the is_mime_markup rule. This rule gives an extra score to pages whose MIME type is specified in the uri_search_mime configuration parameter in section 2.2.3.
change_history	real	10.0	Specifies the weight for the change history rule. This rule scores on the basis of HTTP header "last-modified" value over time, as described in [RFC2616]. Documents that change frequently have a higher score than documents that change less frequently.
sitemap	real	10.0	Specifies the weight for the sitemap rule. The score for the sitemap rule is specified in 2.2.4.26.

2.2.4.26 sitemap_weights Section

In this section, <URL> entries in a sitemap, as specified in [SITEMAPS], can contain a **changefreq** element, which specifies how frequently a URI can be modified. Valid string values for this element are: "always", "hourly", "daily", "weekly", "monthly", "yearly", and "never". The string values are converted into a numerical weight for adaptive crawling.

The **sitemap_weights** section specifies a mapping of the string values to a numerical weight. This numerical weight is used to calculate the score to the sitemap score in the **weights** section.

The adaptive crawling score for a URI is calculated by multiplying the numerical weight by the **sitemap** configuration parameter weight.

The following table specifies the **attrib** elements for this section. The range of these elements MUST be from 0.0 to 1.0.

Name	Type	Default	Meaning
always	real	1.0	Specifies the weight of the **changefreq** value "always" as a numerical value.
hourly	real	0.64	Specifies the weight of the **changefreq** value "hourly" as a numerical value.
daily	real	0.32	Specifies the weight of the **changefreq** value "daily" as a numerical value.
weekly	real	0.16	Specifies the weight of the **changefreq** value "weekly" as a numerical value.
monthly	real	0.08	Specifies the weight of the **changefreq** value "monthly" as a numerical value.
yearly	real	0.04	Specifies the weight of the **changefreq** value "yearly" as a numerical value.
never	real	0.0	Specifies the weight of the **changefreq** value "never" as a numerical value.
default	real	0.16	Specifies the weight for all URIs that are not associated with a **changefreq** value.

2.2.4.27 site_clusters Section

This section specifies configuration parameters that override the normal **crawl routing** of sites within a node scheduler. This parameter ensures that a group of sites is routed to the same node scheduler. The following table specifies the **attrib** elements for this section. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column.

Name	Type	Default	Meaning
The **name** XML attribute specifies a unique identification for this group of sites.	list-string	N/A	Specifies a list of domains that MUST be aggregated to a node scheduler.

2.2.4.28 crawlmode Section

The **crawlmode** section limits the span of a crawl collection. The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
mode	string	FULL	Specifies the depth of the crawling. Valid values are "FULL" or "DEPTH:#", where # is the number of page hops from a start URI.
fwdlinks	boolean	yes	Specifies whether to follow hyperlinks that are located on a different domain.
fwdredirects	boolean	no	Specifies whether to follow external HTTP redirects received from servers. External redirects are HTTP redirects that point from one domain to another domain.
reset_level	boolean	yes	Specifies whether to reset the page hop counter when following a hyperlink to another domain.

2.2.4.29 post_payload Section

The **post_payload** section MUST be used to submit data to **HTTP POST** requests. The data is submitted to URIs that match an URI prefix or that match an exact URI match. The following table specifies the **attrib** XML elements for this section. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column.

Name	Type	Default	Meaning
The **name** attribute is used to match URIs. The section requires an exact match if the **name** XML attribute specifies a URI. To specify a URI prefix, the label "prefix: " MUST be used. Then the leading portion of a URI specifies the remainder of the match.	string	N/A	Specifies the payload data string. This string is posted to URIs that matches a URI or prefix set by the **name** XML attribute.

2.2.4.30 rss Section

The **rss** section initializes and configures **RSS channel** support within a crawl collection. Available **attrib** elements for this section are specified in the following table.

Name	Type	Default	Meaning
start_uris	list-string	N/A	Specifies a list of start URIs that contain RSS channel documents.
start_uri_files	list-string	N/A	Specifies a list of paths to files that contain RSS channel documents. The format of these files MUST be a plain text file with one URI per line.
auto_discover	boolean	no	Specifies whether the Web crawler MUST identify new RSS channels.
follow_links	boolean	yes	Specifies that the Web crawler MUST follow hyperlinks from HTML documents found in the RSS channel.
ignore_rules	boolean	no	Specifies that the Web crawler MUST crawl all documents referenced by the RSS channel, regardless of their inclusion in the include and exclude rules, as specified in sections 2.2.4.1, 2.2.4.2, 2.2.4.3, and 2.2.4.4.

Name	Type	Default	Meaning
index_feed	boolean	no	Specifies whether the Web crawler MUST send RSS channel documents to the indexing engine.
del_expired_links	boolean	no	Specifies whether the Web crawler MUST delete documents from the RSS channel when they expire.
max_link_age	integer	0	Specifies the maximum age, in minutes, for a web document found in an RSS channel. Expired documents will be deleted if the **del_expired_links** configuration parameter is set to yes.
max_link_count	integer	128	Specifies the maximum number of hyperlinks the Web crawler saves for an RSS channel. If the Web crawler encounters more hyperlinks, they expire in a first-in-first-out order. Expired documents will be deleted if **del_expired_links** configuration parameter is set to yes.

2.2.4.31 logins Section

This section MUST specify at least one **logins section** element for HTML form-based authentication. These are associated with specific site logins, each of which MUST contain a unique login name in the **name** attribute. **CT_Login** elements that are specified in section 2.2.6 can be used as an alternative to the **logins** section. The following table specifies the **attrib** elements for a site login section or **CT_Login** element.

Name	Type	Default	Meaning
preload	string	N/A	Specifies the full URI of the page to retrieve previous to processing the login form.
scheme	string	N/A	Specifies the URI scheme of the login site. Valid values: "http" or "https".
site	string	N/A	Specifies the **hostname** of the login form page.
form	string	N/A	Specifies the path to the login form.
action	string	N/A	Specifies whether the form uses HTTP POST or HTTP GET as specified in [RFC2616]. Valid values are: "GET" or "POST".
sites	list-string	N/A	Specifies a list of sites or hostnames that the Web crawler MUST log into previous to beginning the crawl process.
ttl	integer	N/A	Specifies the time, in seconds, that can elapse previous to requiring another login to continue the crawl.
html_form	string	N/A	Specifies the URI to the HTML page containing the login form.
autofill	boolean	N/A	Specifies whether the Web crawler should try to automatically fill out the HTML login form. The **html_form** configuration parameter MUST be specified if this attribute is set to "yes".
relogin_if_failed	boolean	N/A	Specifies whether the Web crawler can attempt to re-login to the crawl site after **ttl** seconds if the login failed.

2.2.4.32 parameters Section

This section sets the authentication credentials used in a HTML form. It MUST be specified within a site **login** section, as specified in section 2.2.4.31, or within a **CT_Login** element, as specified in section 2.2.6. The credential parameters are typically different from HTML form to HTML form.

If the **autofill** configuration parameter is enabled, only variables that are visible in the browser are specified, for example, username and password or their equivalents. In this case the Web crawler MUST retrieve the HTML page and read any **hidden** variables that are required to submit the form. A variable value that is specified in the configuration parameters MUST override any value that was stored in the form. Because there are multiple values for the **name** attribute, a description of each purpose is included in the name column.

The following table specifies the **attrib** elements for this section.

Name	Type	Default	Meaning
The **name** attribute contains the variable of the HTML form to set.	string	N/A	Specifies the values of the HTML form variable.

2.2.4.33 subdomains Section

This section specifies the configuration of a crawl subcollection. The **subdomains** section MUST contain at least one section XML element, each of which specifies a crawl subcollection. A crawl subcollection section MUST contain a unique name by setting the **name** attribute.

Instead of a **subdomains** section a **SubDomain** element can be used as specified in section 2.2.5. If **CT_SubDomain** is used in a crawl collection, the Web crawler will add it to the **subdomains** section. In this case, the **name** attribute is used to create a crawl subcollection section within the **subdomains** section.

Include and exclude rules MUST be specified to limit the scope of a crawl subcollection. These include/exclude rules are: **include_domains**, **exclude_domains**, **include_uris**, and **exclude_uris**, as specified in sections 2.2.4.1, 2.2.4.2, 2.2.4.3 and 2.2.4.4, respectively.

Only a sub-set of the configuration parameters specified in section 2.2.3 can be used within a sub-section. These configuration parameters are the following:

- **accept_compression**
- **allowed_schemes**
- **crawlmode**, **cut_off**
- **delay**
- **ftp_passive**
- **headers**
- **max_doc**
- **proxy**
- **refresh**, **refresh_mode**
- **start_uri_files**

- **start_uris**
- **use_http_1_1**
- **use_javascript**
- **use_sitemaps**

The **refresh** configuration parameters of a crawl subcollection MUST be set lower than the refresh rate of the main crawl collection. The **use_javascript** and **max_doc** configuration parameters MUST NOT obey the **include_uris** and **exclude_uris**.

In addition, the **rss** section and the **variable_delay** section can be used within a crawl subcollection. These are specified in sections 2.2.4.30 and 2.2.4.23, respectively.

2.2.5 CT_SubDomain

This complex type referenced by **CT_DomainSpecification** specifies the configuration of crawl subcollections, as follows.

```
<xs:complexType name="CT_SubDomain">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="attrib" type="CT_attrib"/>
    <xs:element name="section" type="CT_section"/>
  </xs:choice>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
```

A crawl subcollection is an object that differentiates crawl collection members from each other by their definitions. Although its presence is optional, a **CT_SubDomain** element MUST be used to specify a crawl subcollection. A crawl collection can contain multiple **SubDomain** elements. Configuration parameters for a **CT_SubDomain** element are specified in section 2.2.4.33.

attrib: A **CT_attrib** element specifying a configuration parameter.

section: A **CT_section** element specifying a set of configuration parameters and, or sections.

The attribute of **CT_SubDomain** is described in the following table.

Attribute	Type	Meaning
name	**xs:string**	A string specifying the name of the crawl subcollection.

2.2.6 CT_Login

This complex type referenced by **CT_DomainSpecification** MUST be used for HTML forms authentication, as follows.

```
<xs:complexType name="CT_Login">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="attrib" type="CT_attrib"/>
    <xs:element name="section" type="CT_section"/>
  </xs:choice>
```

```
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType
```

Configuration parameters that MUST be set within a **CT_Login** type are specified in section 2.2.4.31. Although the presence of a **Login** element in a crawl collection is optional, a crawl collection can contain multiple **Login** elements.

attrib: A **CT_attrib** element specifying a configuration parameter.

section: A **CT_section** element specifying a set of configuration parameters and, or sections.

The attribute of **CT_Login** is described in the following table.

XML attribute	Type	Meaning
name	**xs:string**	A string specifying the name of the login specification.

2.2.7 CT_Node

Using this complex type referenced by **CT_DomainSpecification**, it is possible to override configuration parameters within a crawl collection or a crawl subcollection for a particular node scheduler by including local parameters within a **CT_Node** element, as follows.

```
  <xs:complexType name="CT_Node">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attrib" type="CT_attrib"/>
      <xs:element name="section" type="CT_section"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType
```

Configuration parameters for this element are specified in sections 2.2.5, 2.2.6, 2.2.3, and 2.2.4.

attrib: A **CT_attrib** element specifying a configuration parameter.

section: A **CT_section** element specifying a set of configuration parameters and, or sections.

The attribute of **CT_Node** is described in the following table.

XML attribute	Type	Meaning
name	**xs:string**	A string specifying the node scheduler identifier to which to apply these configuration parameters.

2.3 Simple Types

The following are simple type specifications.

2.3.1 ST_member

This simple type referenced by **CT_attrib** specifies a string value within a **CT_attrib** type, as specified in section 2.2.3, when the type XML attribute is set to the value **list-string**, as follows.

```
  <xs:simpleType name="ST_member">
    <xs:restriction base="xs:string"></xs:restriction>
  </xs: Defined attrib XML element values
```

A **member** element MUST contain a string that represents a string value of the **list-string**.

2.3.2 ST_type

The value of this enumeration referenced by **CT_attrib** represents the data type of a configuration parameter, as follows.

```
  <xs:simpleType name="ST_type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="boolean"/>
      <xs:enumeration value="string"/>
      <xs:enumeration value="integer"/>
      <xs:enumeration value="list-string"/>
      <xs:enumeration value="real"/>
    </xs:restriction>
  </xs:simpleType
```

Possible values are described in the following table.

Value	Meaning
boolean	Specifies that this attribute is a Boolean configuration parameter. This parameter has two string values; the value "yes" for **true** or "no" for **false**.
integer	Specifies that this attribute is of type INT32, as specified in [MS-DTYP].
real	Specifies that this attribute is of type DOUBLE, as specified in [MS-DTYP].
string	Specifies that this attribute is a string configuration parameter.
list-string	Specifies that this attribute is a list of strings.

3 Structure Examples

The following are configuration examples.

3.1 Simple Configuration

This example configures a simple Web crawler configuration. It is configured to crawl only the contoso.com Web site.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
    <DomainSpecification name="default_example">
        <section name="crawlmode">
            <attrib name="fwdlinks" type="boolean"> no </attrib>
            <attrib name="fwdredirects" type="boolean"> no </attrib>
            <attrib name="mode" type="string"> FULL </attrib>
            <attrib name="reset_level" type="boolean"> no </attrib>
        </section>
        <attrib name="start_uris" type="list-string">
            <member> http://www.contoso.com </member>
        </attrib>
    </DomainSpecification>
</CrawlerConfig>
```

3.2 Typical Configuration

This example crawler configuration contains some of the most common configuration parameters.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
    <DomainSpecification name="default_example">
        <attrib name="accept_compression" type="boolean"> yes </attrib>
        <attrib name="allowed_schemes" type="list-string">
            <member> http </member>
            <member> https </member>
        </attrib>
        <attrib name="allowed_types" type="list-string">
            <member> text/html </member>
            <member> text/plain </member>
        </attrib>
        <section name="cachesize">
            <attrib name="aliases" type="integer"> 1048576 </attrib>
            <attrib name="pp" type="integer"> 1048576 </attrib>
            <attrib name="pp_pending" type="integer"> 131072 </attrib>
            <attrib name="routetab" type="integer"> 1048576 </attrib>
        </section>
        <attrib name="check_meta_robots" type="boolean"> yes </attrib>
        <attrib name="cookie_timeout" type="integer"> 900 </attrib>
        <section name="crawlmode">
            <attrib name="fwdlinks" type="boolean"> yes </attrib>
            <attrib name="fwdredirects" type="boolean"> yes </attrib>
            <attrib name="mode" type="string"> FULL </attrib>
            <attrib name="reset_level" type="boolean"> no </attrib>
        </section>
        <attrib name="csum_cut_off" type="integer"> 0 </attrib>
        <attrib name="cut_off" type="integer"> 5000000 </attrib>
        <attrib name="dbswitch" type="integer"> 5 </attrib>
        <attrib name="dbswitch_delete" type="boolean"> no </attrib>
        <attrib name="delay" type="real"> 60.0 </attrib>
        <attrib name="domain_clustering" type="boolean"> no </attrib>
        <attrib name="enforce_delay_per_ip" type="boolean"> yes </attrib>
        <attrib name="exclude_exts" type="list-string">
            <member> .jpg </member>
            <member> .jpeg </member>
            <member> .ico </member>
            <member> .tif </member>
            <member> .png </member>
            <member> .bmp </member>
            <member> .gif </member>
            <member> .wmf </member>
            <member> .avi </member>
            <member> .mpg </member>
            <member> .wmv </member>
            <member> .wma </member>
            <member> .ram </member>
            <member> .asx </member>
            <member> .asf </member>
            <member> .mp3 </member>
            <member> .wav </member>
            <member> .ogg </member>
            <member> .ra </member>
            <member> .aac </member>
            <member> .m4a </member>
            <member> .zip </member>
            <member> .gz </member>
            <member> .vmarc </member>
            <member> .z </member>
            <member> .tar </member>
            <member> .iso </member>
            <member> .img </member>
            <member> .rpm </member>
            <member> .cab </member>
            <member> .rar </member>
            <member> .ace </member>
            <member> .hqx </member>
            <member> .swf </member>
            <member> .exe </member>
            <member> .java </member>
            <member> .jar </member>
            <member> .prz </member>
            <member> .wrl </member>
            <member> .midr </member>
            <member> .css </member>
            <member> .ps </member>
            <member> .ttf </member>
            <member> .mso </member>
            <member> .dvi </member>
        </attrib>
        <attrib name="extract_links_from_dupes" type="boolean"> no </attrib>
        <attrib name="fetch_timeout" type="integer"> 300 </attrib>
        <attrib name="force_mimetype_detection" type="boolean"> no </attrib>
        <section name="ftp_errors">
            <attrib name="4xx" type="string"> DELETE:3 </attrib>
            <attrib name="550" type="string"> DELETE:0 </attrib>
            <attrib name="5xx" type="string"> DELETE:3 </attrib>
            <attrib name="int" type="string"> KEEP:0 </attrib>
            <attrib name="net" type="string"> DELETE:3, RETRY:1 </attrib>
            <attrib name="ttl" type="string"> DELETE:3 </attrib>
        </section>
        <attrib name="headers" type="list-string">
            <member> User-Agent: FAST Search Web Crawler </member>
        </attrib>
        <attrib name="html_redir_is_redir" type="boolean"> yes </attrib>
        <attrib name="html_redir_thresh" type="integer"> 3 </attrib>
        <section name="http_errors">
            <attrib name="4xx" type="string"> DELETE:0 </attrib>
            <attrib name="5xx" type="string"> DELETE:10 </attrib>
            <attrib name="int" type="string"> KEEP:0 </attrib>
            <attrib name="net" type="string"> DELETE:3, RETRY:1 </attrib>
            <attrib name="ttl" type="string"> DELETE:3 </attrib>
        </section>
        <attrib name="if_modified_since" type="boolean"> yes </attrib>
        <attrib name="javascript_keep_html" type="boolean"> no </attrib>
        <section name="limits">
            <attrib name="disk_free" type="integer"> 0 </attrib>
            <attrib name="disk_free_slack" type="integer"> 3 </attrib>
            <attrib name="max_doc" type="integer"> 0 </attrib>
            <attrib name="max_doc_slack" type="integer"> 1000 </attrib>
        </section>
        <section name="link_extraction">
            <attrib name="a" type="boolean"> yes </attrib>
            <attrib name="action" type="boolean"> yes </attrib>
            <attrib name="area" type="boolean"> yes </attrib>
            <attrib name="card" type="boolean"> yes </attrib>
            <attrib name="comment" type="boolean"> no </attrib>
            <attrib name="embed" type="boolean"> no </attrib>
            <attrib name="frame" type="boolean"> yes </attrib>
            <attrib name="go" type="boolean"> yes </attrib>
            <attrib name="img" type="boolean"> no </attrib>
            <attrib name="layer" type="boolean"> yes </attrib>
            <attrib name="link" type="boolean"> yes </attrib>
            <attrib name="meta" type="boolean"> yes </attrib>
            <attrib name="meta_refresh" type="boolean"> yes </attrib>
        </section>
        <section name="log">
            <attrib name="dsfeed" type="string"> text </attrib>
            <attrib name="fetch" type="string"> text </attrib>
            <attrib name="postprocess" type="string"> text </attrib>
            <attrib name="site" type="string"> text </attrib>
        </section>
        <attrib name="login_failed_ignore" type="boolean"> no </attrib>
        <attrib name="login_timeout" type="integer"> 300 </attrib>
        <attrib name="max_backoff_counter" type="integer"> 50 </attrib>
        <attrib name="max_backoff_delay" type="integer"> 600 </attrib>
        <attrib name="max_doc" type="integer"> 1000000 </attrib>
        <attrib name="max_pending" type="integer"> 2 </attrib>
        <attrib name="max_redirects" type="integer"> 10 </attrib>
        <attrib name="max_sites" type="integer"> 128 </attrib>
        <attrib name="max_uri_recursion" type="integer"> 5 </attrib>
        <attrib name="mufilter" type="integer"> 0 </attrib>
        <attrib name="near_duplicate_detection" type="boolean"> no </attrib>
        <attrib name="obey_robots_delay" type="boolean"> no </attrib>
        <section name="pp">
```

```
            <attrib name="ds_max_ecl" type="integer"> 10 </attrib>
            <attrib name="ds_meta_info" type="list-string">
                <member> duplicates </member>
                <member> redirects </member>
                <member> mirrors </member>
                <member> metadata </member>
            </attrib>
            <attrib name="ds_paused" type="boolean"> no </attrib>
            <attrib name="ds_send_links" type="boolean"> no </attrib>
            <attrib name="max_dupes" type="integer"> 10 </attrib>
            <attrib name="stripe" type="integer"> 1 </attrib>
        </section>
        <section name="ppdup">
            <attrib name="compact" type="boolean"> yes </attrib>
        </section>
        <attrib name="proxy_max_pending" type="integer"> 2147483647 </attrib>
        <attrib name="refresh" type="real"> 1440.0 </attrib>
        <attrib name="refresh_mode" type="string"> scratch </attrib>
        <attrib name="refresh_when_idle" type="boolean"> no </attrib>
        <attrib name="robots" type="boolean"> yes </attrib>
        <attrib name="robots_auth_ignore" type="boolean"> yes </attrib>
        <attrib name="robots_timeout" type="integer"> 300 </attrib>
        <attrib name="robots_tout_ignore" type="boolean"> no </attrib>
        <attrib name="robots_ttl" type="integer"> 86400 </attrib>
        <section name="rss">
            <attrib name="auto_discover" type="boolean"> no </attrib>
            <attrib name="del_expired_links" type="boolean"> no </attrib>
            <attrib name="follow_links" type="boolean"> no </attrib>
            <attrib name="ignore_rules" type="boolean"> no </attrib>
            <attrib name="index_feed" type="boolean"> no </attrib>
            <attrib name="max_link_age" type="integer"> 0 </attrib>
            <attrib name="max_link_count" type="integer"> 128 </attrib>
        </section>
        <attrib name="smfilter" type="integer"> 0 </attrib>
        <attrib name="sort_query_params" type="boolean"> no </attrib>
        <attrib name="start_uris" type="list-string">
            <member> http://www.contoso.com </member>
        </attrib>
        <section name="storage">
            <attrib name="clusters" type="integer"> 8 </attrib>
            <attrib name="compress" type="boolean"> yes </attrib>
            <attrib name="compress_exclude_mime" type="list-string">
                <member> application/x-shockwave-flash </member>
            </attrib>
            <attrib name="datastore" type="string"> bstore </attrib>
            <attrib name="defrag_threshold" type="integer"> 85 </attrib>
            <attrib name="remove_docs" type="boolean"> no </attrib>
            <attrib name="store_dupes" type="boolean"> no </attrib>
            <attrib name="store_http_header" type="boolean"> yes </attrib>
        </section>
        <attrib name="truncate" type="boolean"> no </attrib>
        <attrib name="umlogs" type="boolean"> yes </attrib>
        <attrib name="uri_search_mime" type="list-string">
            <member> text/html </member>
            <member> text/vnd.wap.wml </member>
            <member> text/wml </member>
            <member> text/x-wap.wml </member>
            <member> x-application/wml </member>
            <member> text/x-hdml </member>
        </attrib>
        <attrib name="use_cookies" type="boolean"> no </attrib>
        <attrib name="use_http_1_1" type="boolean"> yes </attrib>
        <attrib name="use_javascript" type="boolean"> no </attrib>
        <attrib name="use_meta_csum" type="boolean"> no </attrib>
        <attrib name="use_sitemaps" type="boolean"> no </attrib>
        <section name="workqueue_priority">
            <attrib name="default" type="integer"> 1 </attrib>
            <attrib name="levels" type="integer"> 1 </attrib>
            <attrib name="pop_scheme" type="string"> default </attrib>
            <attrib name="start_uri_pri" type="integer"> 1 </attrib>
        </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.3 Crawl Subcollection

The following example initializes a subcollection using the **SubDomain** XML element, specified in 2.2.5.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="subcollection_example">
      <SubDomain name="subdomain_1">
         <section name="include_uris">
            <attrib name="prefix" type="list-string">
               <member> http://www.contoso.com/index </member>
            </attrib>
         </section>
         <attrib name="refresh" type="real"> 60.0 </attrib>
         <attrib name="delay" type="real"> 10.0 </attrib>
         <attrib name="start_uris" type="list-string">
            <member> http://www.contoso.com/ </member>
         </attrib>
      </SubDomain>
   </DomainSpecification>
</CrawlerConfig>
```

The following configuration is the same as the previous configuration, with the exception that it uses a **subdomains** section as specified in section 2.2.4.33.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="subcollection_example">
      <section name="subdomains">
         <section name="subdomain_1">
            <section name="include_uris">
               <attrib name="prefix" type="list-string">
                  <member> http://www.contoso.com/index </member>
               </attrib>
            </section>
            <attrib name="refresh" type="real"> 60.0 </attrib>
            <attrib name="delay" type="real"> 10.0 </attrib>
            <attrib name="start_uris" type="list-string">
               <member> http://www.contoso.com/ </member>
            </attrib>
         </section>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.4 Login

This example configures a crawl collection to crawl a site that is associated with an HTML login form. The login parameters are provided by the Login XML element, specified in section 2.2.6.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="login_example">
      <Login name="mytestlogin">
          <attrib name="preload" type="string">http://preload.contoso.com/
          </attrib>
          <attrib name="scheme" type="string"> https </attrib>
          <attrib name="site" type="string"> login.contoso.com  </attrib>
          <attrib name="form" type="string"> /path/to/some/form.cgi </attrib>
          <attrib name="action" type="string">POST</attrib>
          <section name="parameters">
             <attrib name="user" type="string"> username </attrib>
             <attrib name="password" type="string"> password </attrib>
          </section>
          <attrib name="sites" type="list-string">
             <member> site1.contoso.com  </member>
             <member> site2.contoso.com  </member>
          </attrib>
          <attrib name="ttl" type="integer"> 7200 </attrib>
          <attrib name="html_form" type="string">
             http://login.contoso.com/login.html
           </attrib>
           <attrib name="autofill" type="boolean"> yes </attrib>
           <attrib name="relogin_if_failed" type="boolean"> yes </attrib>
      </Login>
   </DomainSpecification>
</CrawlerConfig>
```

3.5 Node

This example stipulates a multiple node installation. One of the node schedulers is named crawler_node1, which is configured with a different **delay** configuration parameter than the other nodes by using the Node XML element, as specified in section 2.2.7.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="node_example ">
      <attrib name="delay" type="real"> 60.0 </attrib>
      <Node name="crawler_node1">
         <attrib name="delay" type="real"> 90.0 </attrib>
      </Node>
   </DomainSpecification>
</CrawlerConfig>
```

3.6 Workqueue

This example configures crawl queues with different priority levels using the **workqueue_priority** section, specified in section 2.2.4.15. In this example two priority levels are created. URIs from web005.contoso.com are given priority level 1; other URIs are inserted into the level 2 crawl queue.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="workqueue_example">
      <section name="workqueue_priority">
         <attrib name="levels" type="integer"> 2 </attrib>
         <attrib name="default" type="integer"> 2 </attrib>
         <attrib name="start_uri_pri" type="integer"> 1 </attrib>
         <attrib name="pop_scheme" type="string"> wrr </attrib>
         <attrib name="put_scheme" type="string"> include </attrib>
         <section name="1">
            <attrib name="share" type="integer"> 10 </attrib>
            <section name="include_domains">
               <attrib name="suffix" type="list-string">
                  <member> web005.contoso.com  </member>
               </attrib>
            </section>
         </section>
         <section name="2">
            <attrib name="share" type="integer"> 5 </attrib>
            <section name="include_domains">
               <attrib name="suffix" type="list-string">
                  <member> web002.contoso.com  </member>
               </attrib>
            </section>
         </section>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.7 Variable Delay

In this example, the Web crawler uses different delay intervals during the week, as specified in section 2.2.4.23. On Wednesday between 9:00 a.m. and 7:00 p.m. the Web crawler uses a delay of 20 seconds. On Monday between 9:00 a.m. and 5:00 p.m. the crawler suspends crawling, and any other time of the week the Web crawler uses a delay of 60 seconds.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="variable_example">
      <attrib name="delay" type="real"> 60.0 </attrib>
      <section name="variable_delay">
         <attrib name="Wed:09-Wed:19" type="string">20 </attrib>
         <attrib name="Mon:09-Mon:17" type="string">suspend</attrib>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.8 HTTP Errors

This example demonstrates how the **http_errors** section, as specified in section 2.2.4.11, handles various HTTP errors with specificity.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="http_errors_example">
      <section name="http_errors">
         <attrib name="408" type="string"> KEEP </attrib>
         <attrib name="4xx" type="string"> DELETE </attrib>
         <attrib name="5xx" type="string"> DELETE:10, RETRY:3 </attrib>
         <attrib name="ttl" type="string"> DELETE:3 </attrib>
         <attrib name="net" type="string"> DELETE:3 </attrib>
         <attrib name="int" type="string"> KEEP </attrib>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.9 Passwd

This example demonstrates how to initialize user credentials using the **passwd** section, as specified in 2.2.4.20.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="password_example">
      <section name="passwd">
         <attrib name="http://www.contoso.com/confidential1/" type="string">
            user:password:contoso:auto
         </attrib>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.10 Site Clustering

This example demonstrates how to cluster a set of sites using the **site_clusters** section, as specified in 2.2.4.27.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig >
   <DomainSpecification name="password_example">
      <section name="site_clusters">
         <attrib name="mycluster" type="list-string">
            <member> site1.constoso.com </member>
            <member> site2.constoso.com </member>
            <member> site3.constoso.com </member>
         </attrib>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.11 Post Payload

This example demonstrates how to initialize an HTTP POST payload for the URI http://www.contoso.com /secure using the **post_payload** section, as specified in section 2.2.4.29.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="post_payload_example">
      <section name="post_payload">
         <attrib name="prefix:http://www.contoso.com/secure" type="string">
variable1=value1&amp;variableB=valueB </attrib>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

3.12 Feeding

This example demonstrates how to initialize feeding destinations using the **feeding** section, as specified in section 2.2.4.9.

```
<?xml version="1.0" encoding="utf-8"?>
<CrawlerConfig>
   <DomainSpecification name="feeding_example">
      <section name="feeding">
         <section name="Global_News">
            <attrib name="collection" type="string"> collection_A </attrib>
            <attrib name="destination" type="string"> default </attrib>
            <attrib name="primary" type="boolean"> yes </attrib>
            <attrib name="paused" type="boolean"> no </attrib>
         </section>
         <section name="Local_News">
            <attrib name="collection" type="string"> collection_B </attrib>
            <attrib name="destination" type="string"> default </attrib>
            <attrib name="primary" type="boolean"> no </attrib>
            <attrib name="paused" type="boolean"> no </attrib>
         </section>
      </section>
   </DomainSpecification>
</CrawlerConfig>
```

4 Security Considerations

None.

5 Appendix A: XML Schema

A Web crawler configuration XML file is formatted in accordance with the following XML schema.

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

  <xs:element name="CrawlerConfig" type="CT_CrawlerConfig"/>

  <xs:complexType name="CT_CrawlerConfig">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="DomainSpecification" type="CT_DomainSpecification"/>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="CT_DomainSpecification">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attrib" type="CT_attrib" maxOccurs="unbounded"/>
      <xs:element name="section" type="CT_section"/>
      <xs:element name="SubDomain" type="CT_SubDomain"/>
      <xs:element name="Login" type="CT_Login"/>
      <xs:element name="Node" type="CT_Node"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_attrib" mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="member" type="ST_member"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="type" type="ST_type" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_section">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="attrib" type="CT_attrib"/>
        <xs:element name="section" type="CT_section"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_SubDomain">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attrib" type="CT_attrib"/>
      <xs:element name="section" type="CT_section"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_Login">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attrib" type="CT_attrib"/>
      <xs:element name="section" type="CT_section"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="CT_Node">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attrib" type="CT_attrib"/>
      <xs:element name="section" type="CT_section"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:simpleType name="ST_type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="boolean"/>
      <xs:enumeration value="string"/>
      <xs:enumeration value="integer"/>
      <xs:enumeration value="list-string"/>
      <xs:enumeration value="real"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_member">
    <xs:restriction base="xs:string"></xs:restriction>
  </xs:simpleType>
</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