

[MS-FSWAADM]: WebAnalyzer Administration and Status Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
11/06/2009	0.1	Major	Initial Availability
02/19/2010	1.0	Minor	Updated 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.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References.....	6
1.2.1	Normative References.....	6
1.2.2	Informative References	7
1.3	Protocol Overview (Synopsis)	7
1.4	Relationship to Other Protocols.....	7
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement.....	8
1.7	Versioning and Capability Negotiation.....	8
1.8	Vendor-Extensible Fields.....	8
1.9	Standards Assignments	8
2	Messages.....	9
2.1	Transport.....	9
2.2	Message Syntax	9
2.2.1	Configuration Options	9
2.2.1.1	Global Configuration Options	9
2.2.1.2	FAST Distributed Make Configuration Options.....	10
2.2.1.3	Web Analyzer View Configuration Options	10
2.2.2	Status Structures	11
2.2.2.1	Global Status Structure	11
2.2.2.2	View Status Structure	12
2.2.3	Nested Status Arrays and Structures	13
2.2.3.1	Crawl Collection Status Structure.....	13
2.2.3.2	system_status Array	14
2.2.3.3	processing_status Array	14
2.2.3.4	run_stats Structure.....	15
2.2.4	Web Analyzer View and Crawl Collection Mappings	15
2.2.4.1	allviews Structure	15
2.2.4.2	Web Analyzer Views Array	15
2.2.4.3	Preferred Web Analyzer View Structure	16
2.2.5	Analysis Stages.....	16
2.2.5.1	Analysis Stages	16
2.2.5.2	Analysis Substages	16
2.2.6	URL Relevance Structure	17
2.2.6.1	anchors Array	17
2.2.7	Log Levels	17
2.2.8	Error Handling	18
2.2.9	WebAnalyzer Management Methods.....	18
2.2.9.1	CreateView	18
2.2.9.2	DeleteCollection	19
2.2.9.3	DeleteView	19
2.2.10	Configuration Methods	19
2.2.10.1	GetConfig	19
2.2.10.2	GetFDMConfig	19
2.2.10.3	GetLogLevel	20
2.2.10.4	GetViewConfig	20
2.2.10.5	GetViewCurrentRunConfig	20
2.2.10.6	SetConfig.....	20

2.2.10.7	SetFDMConfig.....	21
2.2.10.8	SetLogLevel	21
2.2.10.9	SetViewConfig	21
2.2.11	Status Methods	22
2.2.11.1	GetStatus	22
2.2.11.2	GetURIRelevanceData	22
2.2.11.3	GetViewStatus.....	22
2.2.12	Processing Management Methods	23
2.2.12.1	ForceProcessing.....	23
2.2.12.2	PauseProcessing	23
2.2.12.3	ProcessOnce.....	23
2.2.12.4	StartProcessing	24
2.2.12.5	StopProcessing	24
2.2.13	WebAnalyzer Data Management Methods	24
2.2.13.1	SetPreferredView	25
2.2.13.2	UnsetPreferredView.....	25
2.2.13.3	GetPreferredViews	25
2.2.13.4	AddCollections.....	25
2.2.13.5	RemoveCollections.....	26
2.2.13.6	AlterDescription	26
2.2.13.7	GetAllViews.....	26
3	Protocol Details.....	27
3.1	Common Details	27
3.1.1	Abstract Data Model	27
3.1.2	Timers	27
3.1.3	Initialization	27
3.1.4	Higher-Layer Triggered Events.....	27
3.1.5	Message Processing Events and Sequencing Rules.....	27
3.1.6	Timer Events	27
3.1.7	Other Local Events	27
3.2	Protocol Client Details	27
3.2.1	Abstract Data Model	27
3.2.2	Timers	27
3.2.3	Initialization	27
3.2.4	Higher-Layer Triggered Events.....	28
3.2.5	Message Processing Events and Sequencing Rules.....	28
3.2.6	Timer Events	28
3.2.7	Other Local Events	28
3.3	Protocol Server Details	28
3.3.1	Abstract Data Model	28
3.3.2	Timers	28
3.3.3	Initialization	28
3.3.4	Higher-Layer Triggered Events.....	28
3.3.5	Message Processing Events and Sequencing Rules.....	29
3.3.5.1	AddCollections.....	29
3.3.5.2	ConfigurationChanged	29
3.3.5.3	CreateView	29
3.3.5.4	DeleteCollection	29
3.3.5.5	DeleteView	29
3.3.5.6	RemoveCollections.....	29
3.3.5.7	SetPreferredView.....	29
3.3.5.8	UnsetPreferredView	30

3.3.6	Timer Events	30
3.3.7	Other Local Events	30
4	Protocol Examples.....	31
4.1	GetAllViews Method	31
4.2	StartProcessing Method	32
5	Security.....	33
5.1	Security Considerations for Implementers.....	33
5.2	Index of Security Parameters	33
6	Appendix A: Product Behavior.....	34
7	Change Tracking.....	35
8	Index	36

1 Introduction

This document specifies the WebAnalyzer Administration and Status Protocol, which transmits status and configuration options between a protocol client and a protocol server. It enables the protocol client to query the protocol server for status information.

1.1 Glossary

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

Augmented Backus-Naur Form (ABNF)
Coordinated Universal Time (UTC)
UTF-8

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

anchor text
base port
crawl collection
document identifier
equivalence class
hyperlink
search index
Web analyzer
Web analyzer view
Web site

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

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.

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004,
<http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30>

Note There is a charge to download the specification.

[MS-FSCDCFG] Microsoft Corporation, "[Component Distribution Configuration File Format Specification](#)"

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

[MS-FSWASDS] Microsoft Corporation, "[WebAnalyzer/SPRel Data Serving Protocol Specification](#)"

[MS-FSXTAPI] Microsoft Corporation, "[XML-RPC Translatable API Structure 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>

[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>

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

[XML-RPC] Winer, D., "XML-RPC Specification", June 1999, <http://www.xmlrpc.com/spec>

1.2.2 Informative References

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

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

1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to issue requests to the **hyperlink** analysis application protocol server, or **Web analyzer**. The requests can be queries for status of different types, to create or delete Web analyzer views or to manage how the protocol server analyzes hyperlinks.

The protocol is a Remote Procedure Call (RPC) that uses XML to encode the methods and responses, and HTTP as a transport mechanism.

Communications consist of the following:

1. The application running on the protocol client issues a request to the protocol server.
2. The protocol server returns a response to the protocol client.

The protocol server does not initiate communication with the protocol client. The protocol client is aware of the hostname and port of the protocol server.

1.4 Relationship to Other Protocols

The WebAnalyzer Administration and Status Protocol uses XML-RPC over HTTP as shown in the following layering diagram:

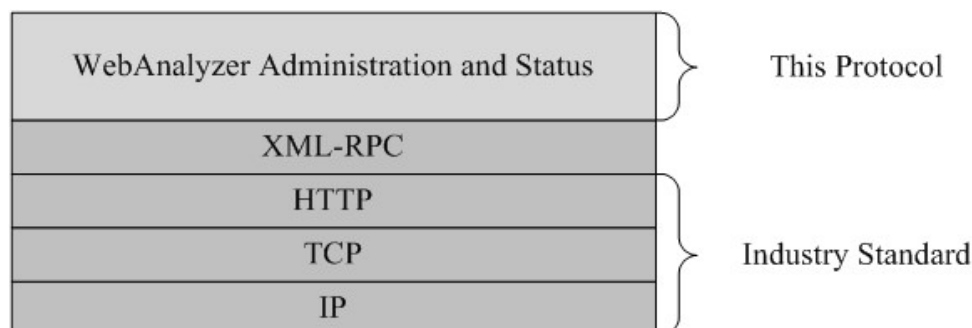


Figure 1: This protocol in relation to other protocols

For more information, see [\[XML-RPC\]](#).

1.5 Prerequisites/Preconditions

It is assumed that the protocol client has obtained the host name and the port for the protocol server before the protocol is initiated.

The protocol requires that the protocol client MUST initiate the setup of a TCP connection between the protocol client and the protocol server. The port number for the connection MUST be **base port** plus 300.

1.6 Applicability Statement

This protocol is used in a distributed system in which protocol clients communicates with a hyperlink analysis engine application to get status, change configuration and manage when analysis should be run.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses the transport protocol specified in [\[XML-RPC\]](#). The syntax used to specify the XML-RPC methods in this specification is translated to xml as specified in [\[MS-FSXTAPI\]](#).

2.2 Message Syntax

The format of the HTTP body requests and responses is specified in [\[XML-RPC\]](#). The HTTP POST path, as specified in [\[RFC2616\]](#), contains the value "/RPC2". The protocol server and the protocol client MUST support both HTTP version 1.0 and HTTP version 1.1.

Implementers MUST encode the following data types as specified in [\[XML-RPC\]](#):

- **array**
- **boolean**
- **double**
- **int**
- **string**
- **struct**

The protocol also specifies a **dynamic** type, which implementers use to define arguments that can be either **boolean**, **double**, **int** or **string**. The size of the dynamic type varies based on which [\[XML-RPC\]](#) type the dynamic type represents. All strings use **UTF-8** encoding.

Some messages contain **double** fields that are encoded as **string** fields. Such a value consists of an ASCII string that specified by the following **Augmented Backus-Naur Form (ABNF)** rules, as specified in [\[RFC5234\]](#).

```
floatnumber = pointfloat / exponentfloat
pointfloat  = ([intpart] fraction) / (intpart ".")
exponentfloat = intpart / pointfloat exponent
intpart     = <1>*<17>DIGIT
fraction    = "." <1>*<16>DIGIT
exponent    = "e" ["-"] <1>*<3>DIGIT
```

2.2.1 Configuration Options

This section specifies all the supported configuration options.

2.2.1.1 Global Configuration Options

The following table specifies all the configuration options that are global for the protocol server.

Option name	Type	Description
batch_size	int	Specifies the number of operations that can be submitted in a batch to the indexing component.

Option name	Type	Description
callback_timeout	int	Specifies the maximum number of seconds the analysis nodes can wait for a response after sending the final batch of operations to the indexing component.
concurrent_feeds	int	Specifies the number of processes that can simultaneously send operations to the indexing component. If the value of the <i>cpus</i> configuration option is less than that of the <i>concurrent_feeds</i> field, the <i>cpus</i> option specifies the number of processes used.
cpus	int	Specifies the number of operations that can simultaneously run on each analysis node in the system.
pollwalr_callback_interval	int	Specifies the number of seconds that will elapse before the protocol server checks for new information.

2.2.1.2 FAST Distributed Make Configuration Options

The following table specifies all the configuration options .

Option name	Type	Description
disk_free	int	This option specifies the minimum amount of available disk space on each node. This field is global, and any node that has a smaller amount of available disk space than this limit MUST NOT participate in the analysis. This field is specified in megabytes and MUST NOT be less than 1.
verbose	int	If true , verbose logging is enabled for the analysis framework. This log contains only information about the analysis; it is not specified by the verbose log level that is specified in section 2.2.7 . The value of this option is 0 to mean false or 1 to mean true .

2.2.1.3 Web Analyzer View Configuration Options

The following table specifies all configuration options that are valid for a **Web analyzer view**. Each Web analyzer view has its own set of options.

Option name	Type	Description
drop_intra	int	If true , all the hyperlinks that link from one document to another on the same Web site (1) are discarded. If true , the value MUST be equal to 1. If false , the value MUST be equal to 0.
force_collections	int	If true , all the crawl collections that are associated with the Web analyzer view are updated. If false , the existing hyperlinks are used. If true , the value MUST be equal to 1. If false , the value MUST be equal to 0.
run_partial_update	int	If true , partial update operations are submitted for all the documents that have changed and that exist in the search index . If true , the value MUST be equal to 1. If false , the value MUST be equal to 0.
sort_buffer	int	This specifies the maximum amount of memory, in megabytes, that can be used to sort data. This value is specified on a per task basis. If several tasks are running on the same protocol server, the total amount of memory used, can be greater than this value.

2.2.2 Status Structures

This section specifies all the *status* structures that are used by the Protocol.

2.2.2.1 Global Status Structure

The following table specifies the content of the structure that is returned by the **GetStatus** method.

Member name	Type	Description
collections	struct	This structure contains all the crawl collections in the system that are ready for analyzing. Every member name represents a crawl collection name. Every member field is a structure as specified in section 2.2.3.1 .
collections_being_created	array	This array contains all the crawl collections that are created. Every element of the array is of type string .
collections_being_deleted	array	This array contains all the crawl collections to delete. Every element of the array is of type string .
end_factor	int	This represents the number of partitions into which the analysis results are divided. It is equal to the number of elements contained in the <i>walookupdb</i> s array.
process_view_queue	array	This ordered array contains all the Web analyzer views that are scheduled for analysis and that are associated with crawl collections that contain unprocessed information. The first Web analyzer view in the array MUST be the next one to be processed. Every element in the array is of type string .
split_factor	int	This represents the number of partitions into which the data is divided.
system_status	array	This array is a <i>system_status</i> array as specified in section 2.2.3.2 .
views	struct	This structure contains all the Web analyzer views in the system that are not being created or deleted. Every member name is the name of a Web analyzer view. Every member is an array as specified in section 2.2.3.3 .
views_being_created	array	This array contains all the Web analyzer views that are created. Every element in the array is of type string .
views_being_deleted	array	This array contains all the Web analyzer views to delete. Every element in the array is of type string .
walookupdb	array	This array contains the Web analyzer lookup database nodes. The array contains one entry for each link processing component as specified in [MS-FSCDCFG] . Each array entry is a two-element array in which the first element is of type string and specifies a host name. The second element is of type int and specifies the database partition that contains the data to locate.
waworkers	array	This array contains the Web analyzer worker nodes. The array contains one entry for each link processing component as specified in [MS-FSCDCFG] . Each array entry is a two-element array in which the first element is of type string specifies a host name and the second element is of type string and specifies a path.

2.2.2.2 View Status Structure

The following table specifies the content of the structure that is returned by the **GetViewStatus** method.

Member name	Type	Description
collections	array	This array contains all the crawl collections that are associated with the Web analyzer view. Every element in the array is of type string .
generation	int	This specifies the number of times that the view has been processed.
last_time_finished	string	This specifies the most recent time that the Web analyzer view analysis process finished processing. It is an integer that specifies the time in seconds that elapsed after 00:00:00 1970-01-01 Coordinated Universal Time (UTC) . It is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57. If the Web analyzer view has not been processed, this field contains a value of 0.
last_time_started	string	This specifies the most recent time that the Web analyzer view analysis process finished processing. The field contains an integer that specifies the time in seconds that elapsed after 00:00:00 1970-01-01 UTC . It is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57. If the Web analyzer view has not been processed, this field contains a value of 0.
number_of_coll_links	struct	This structure contains all the crawl collections that were associated with the Web analyzer view during its most recent processing. Every member contains the number of links that the crawl collection contained during the processing. Both the member names and the member values are of type string .
number_of_sites	string	This value specifies the number of Web sites (1) that were found during the most recent analysis. The value is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
number_of_uris	string	This value specifies the number of URLs that were found during the most recent analysis. The value is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
number_of_view_links	string	This value specifies the number of hyperlinks that were found during the most recent analysis. The value is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
preferred	array	This array contains all the crawled collections that contain relevance information from this Web analyzer view as specified by [MS-FSWASDS] .
processing_status	array	This array contains analysis status information as specified in section 2.2.3.3 .
run_stats	struct	This structure contains run statistics as specified in section 2.2.3.4 .
time_spent	int	If the Web analyzer view is being processed, this value represents the number of seconds that elapsed after the analysis began processing. If the Web analyzer view is not being processed, this value MUST be -1.

2.2.3 Nested Status Arrays and Structures

This section specifies all the arrays and structures that are contained within the top-level status structures as specified in section [2.2.2](#).

2.2.3.1 Crawl Collection Status Structure

This is returned as part of the global status structure as specified in section [2.2.2.1](#). The following table specifies the content of the crawl collection status structure.

Member name	Type	Description
cleared	string	This specifies the date the crawl collection was most recently cleared of content. It is formatted as specified in [ISO-8601] . If the crawl collection has not yet been cleared, then this field is the same as the <i>created</i> field.
created	string	This specifies the date the crawl collection was created. It is formatted as specified in [ISO-8601] .
generation	string	This specifies the number of times that the crawl collection has been processed. It is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
last_preferred_view	string	This value specifies which Web analyzer view the crawl collection used to retrieve relevance information during the most recent processing of the crawl collection. The value is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
last_time_finished	string	This specifies the most recent time that the crawl collection analysis process finished processing. It is an integer that specifies the time in seconds that elapsed after 00:00:00 1970-01-01 UTC. It is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57. If the crawl collection has not been processed, the value MUST be -1.
last_time_started	string	This specifies the most recent time that the crawl collection started an analysis process that finished. It is an integer that specifies the time in seconds that elapsed after 00:00:00 1970-01-01 UTC. It is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57. If the crawl collection has not been processed, the field contains -1.
number_of_links	string	This value specifies the number of hyperlinks that the crawl collection contained when it was most recently processed. The value is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
started	string	This value specifies whether the crawl collection is associated with a Web analyzer view that is currently running. If so, the field contains "True"; if not, the field contains "False". The value is encoded as an ASCII string.
timestamp	string	This specifies the most recent time that the crawl collection received new information. It is an integer that specifies the time in seconds that elapsed after 00:00:00 1970-01-01 UTC. It is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57. If the crawl collection has not been processed, the field contains -1.

2.2.3.2 system_status Array

This is returned as part of the global status structure as specified in section [2.2.2.1](#). The following table specifies the content of the *system_status* array.

Element	Type	Description
0	string	If the protocol server is booting for the first time, this MUST contain the value "Bootstrap". Otherwise, it contains the value "Running".
1	string	If one or more worker processes are not responding, this MUST contain the value "WorkerFailure". If another error occurred, it contains the value "WError". In all other cases, it contains the value "NoError".
2	string	If Element 1 contains "NoError", then this contains the value "No error". Otherwise, this contains a string that specifies the error.

2.2.3.3 processing_status Array

This is returned as part of the Web analyzer view status structure as specified in section [2.2.2.2](#). The following table specifies the content of the *processing_status* array.

Element	Type	Description
0	string	This field specifies the overall analysis status for the Web analyzer view: <ul style="list-style-type: none">▪ If the Web analyzer view is running, the field contains "running".▪ If the Web analyzer view is running but about to stop, the field contains "stopping".▪ If the Web analyzer view is stopped, the field contains "stopped".▪ If the Web analyzer view is running but about to pause, the field contains "pausing".▪ If the Web analyzer view is paused, the field contains "paused".▪ If none of the preceding conditions apply, the field contains "scheduled".
1	int	This specifies the position of the Web analyzer view within the analysis queue. If the Web analyzer view state contains the value "stopped", "paused", or "scheduled" with no new information, the field in this position contains the value -1. If the Web analyzer view state contains the value "running", "pausing", or "stopping", the field contains a value of 0. Web analyzer view states are specified in section 2.2.3.3
2	string	This specifies the stage that the Web analyzer view processing is performing, or the type of hyperlink analysis to perform. For more details, see section 2.2.5.2 . The field is encoded as a string.
3	string	This specifies the substage of the Web analyzer view processing. For more details, see section 2.2.5.2 . The field is encoded as a string.
4	double or int	This specifies the percentage of the current stage that has finished. If the Web analyzer view is not being processed, this field is of type int and contains a value of 0.

2.2.3.4 run_stats Structure

This is returned as part of the *view_status* structure as specified in section [2.2.2.2](#). The *run_stats* structure contains statistics about the amount of time that a Web analyzer view consumed while analyzing links. The following table specifies the content of the *run_stats* structure.

Member name	Type	Description
count	int	This value specifies the number of times that the Web analyzer view has finished the analyzing process.
keeplast	int	This value specifies the number of analyzing runs for which to maintain run times.
lastn	array	This array contains the amount of time consumed during the N most recent analysis runs. The number of elements in the array MUST NOT be greater than the value of the <i>keeplast</i> field. The value of each element is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
max	int	This specifies the number of seconds that the slowest analysis run consumed. If the Web analyzer view has not yet been processed, this MUST contain the value 2147483648.
min	int	This specifies the number of seconds that the fastest analysis run consumed. If the Web analyzer view has not yet been processed, this MUST contain the value 2147483647.
sum	int	This value specifies the total amount of time that was consumed while analyzing hyperlinks.

2.2.4 Web Analyzer View and Crawl Collection Mappings

This section specifies all the structures and arrays that specify mappings between Web analyzer views and crawl collections.

2.2.4.1 allviews Structure

This is returned by the **GetAllViews** method. It contains information about which crawl collections are components of a Web analyzer view. Each member name is a Web analyzer view name, and each member value is an array.

Member name	Type	Description
Web analyzer view name	array	This array contains a description of the Web analyzer view and all associated crawl collections, as specified in section 2.2.4.2 .

2.2.4.2 Web Analyzer Views Array

This is returned in the *allviews* structure as specified in section [2.2.4.1](#).

Element	Type	Description
0	string	A description of the Web analyzer view

Element	Type	Description
1	array	An array of strings that specify the names of crawl collections that are associated with the Web analyzer view.

2.2.4.3 Preferred Web Analyzer View Structure

This is returned by the **GetPreferredViews** method. It contains information about the Web analyzer views from which each crawl collection retrieves relevance information, as specified by [\[MS-FSWASDS\]](#).

Member name	Type	Member value	Type
crawl collection name	string	Web analyzer view name	string

2.2.5 Analysis Stages

The hyperlink analysis that the protocol server performs is divided into stages that specify the type of analysis to perform. Each stage is divided into three substages that specify how much processing was performed. This section specifies the stages and substages.

The following table specifies the name and description of each analysis stage.

Stage name	Description
wacollproc	For each crawl collection, collects any new hyperlinks and updates the existing hyperlinks.
waviewprep	Merges all the hyperlinks from all the crawl collections.
warelevancy	Performs a hyperlink analysis.
walookupdeployer	Deploys the output databases.
wapartialupdate	Updates the search index.
nothing	A placeholder to use only when no analysis is running.

2.2.5.1 Analysis Stages

2.2.5.2 Analysis Substages

The following table specifies the name and description of each analysis substage.

Substage name	Description
processing	This substage is used as the main part of the stage.
postprocessing	This substage is used when postprocessing operations, such as cleanup, are running.
done	This substage is used if the stage is finished and the next one is about to begin. If the stage contains the value nothing , the substage MUST contain the value done .

2.2.6 URL Relevance Structure

The following table specifies the content of the structure that is returned by the **GetURIRelevanceData** method.

Member name	Type	Description
contentid	string	This is the document identifier (3) for the document that is specified by this URL relevance structure.
siterank	string	This value ranks the Web site (1) that is associated with the URL. The field contains a double value that is encoded as a string . The encoding is performed as specified in section 2.2 .
anchors	array	This array of arrays contains all the anchor text strings that are used in hyperlinks that point to the URL. For more details, see section 2.2.6.1 .
rank	string	This value ranks the URL. The field contains a double value that is encoded as a string . The encoding is performed as specified in section 2.2 .
urieqs	array	This array specifies the equivalence class for the URL. Each element in the array is of type string .

2.2.6.1 anchors Array

This is associated with one specific URL and comprises one or more arrays whose format is specified in the following table.

Element	Type	Description
0	string	The anchor text in a hyperlink that refers to the URL.
1	string	The number of times that the anchor text has referred to the URL. This value is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
2	string	The weight of the anchor text, based on the number of times that it referred to the URL and also based on the rank of the referring hyperlinks. This is a double field that is encoded as a string , as specified in section 2.2 .
3	string	The number of times that the anchor text occurred in a hyperlink that was found during the analysis. This field is encoded as an ASCII string, so each digit is represented by a byte that contains a number from 48 through 57.
4	string	The weight of the anchor text, based on the number of times that it referred to any URL and also based on the rank of the referring hyperlinks. This is a double field that is encoded as a string , as specified in section 2.2 .

2.2.7 Log Levels

Log level codes are specified in the following table.

Log level name	Description
error	ERROR and CRITICAL log messages

Log level name	Description
warning	The preceding level plus WARNING messages
info	The preceding level plus INFO messages
verbose	The preceding level plus VERBOSE messages
debug	The preceding level plus DEBUG messages

2.2.8 Error Handling

The XML-RPC protocol supports a special message, known as a fault, to report errors back to the protocol client. The fault **MUST** contain a fault code and a fault string as specified in [\[XML-RPC\]](#).

Most errors that occur generate faults. Whenever a method generates a fault, it substitutes the fault for the return value of the method. The return values that are specified in the following sections apply only to successful calls; every method **MUST** return a fault if the call is unsuccessful.

The fault code is of type **int** and contains a value of 1. The fault string is specified with the following Augmented Backus-Naur Form (ABNF) rules:

```
errormsg = prefix type delim errortxt

delim = %d39.38.103.116.59.58.32.32.60
prefix = %d38.108.116.59.116.121.112.101.32.39

type = %d101.120.99.101.112.116.105.111.110.46 (exception / attributeerror) %d46

exception = %d69.120.99.101.112.116.105.111.110
attributeerror = %d65.116.116.114.105.98.117.116.101.69.114.114.111.114
errortxt = 1*(VCHAR / SP)
```

exception: This is the fault message if the fault occurred within the method that was called.

attributeerror: This is the fault message when an unknown method is called by the method or calling application.

2.2.9 WebAnalyzer Management Methods

Every method that is specified in this section **MUST** use all parameters.

2.2.9.1 CreateView

This creates a new Web analyzer view.

```
int CreateView(string View, string Description)
```

View: The name of the Web analyzer view.

Description: A description of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.9.2 DeleteCollection

This deletes an existing crawl collection.

```
int DeleteCollection(string Collection)
```

Collection: The name of the crawl collection.

Return value	Description
1	The protocol server MUST return 1.

2.2.9.3 DeleteView

This deletes an existing Web analyzer view.

```
int DeleteView(string View)
```

View: The name of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.10 Configuration Methods

Every method that is specified in this section MUST use all parameters.

2.2.10.1 GetConfig

This returns all the global configuration options that are used by the protocol server.

```
struct GetConfig()
```

Return value	Description
A <i>global_configuration</i> structure	The protocol server MUST return a structure that contains all the global configuration options as specified in section 2.2.1.1 .

2.2.10.2 GetFDMConfig

This returns all the configuration options that are used by the protocol server to control the analysis.

```
struct GetFDMConfig(string Master)
```

Master: A string that contain the value "proc".

Return value	Description
A <i>fdm_configuration</i> structure	The protocol server MUST return a structure that contains all configuration options as specified in section 2.2.1.2 .

2.2.10.3 GetLogLevel

This queries the protocol server for its current log verbosity level.

```
string GetLogLevel()
```

Return value	Description
The log level	The protocol server MUST return a mask of one or more log levels as specified in section 2.2.7 .

2.2.10.4 GetViewConfig

This returns all the configuration options that are used by the specific Web analyzer view. The configuration options MUST specify the most-recent settings.

```
struct GetViewConfig(string View)
```

View: The name of the Web analyzer view.

Return value	Description
A <i>view_configuration</i> structure	The protocol server MUST return a structure that contains all the configuration options for the Web analyzer view as specified in section 2.2.1.3 .

2.2.10.5 GetViewCurrentRunConfig

This returns all the configuration options for the specified Web analyzer view. If the Web analyzer view is being processed, the configuration options returned by this method are the ones in use. Any changes to the configuration options after the analysis began processing do not need to be reflected by this method.

```
struct GetViewCurrentConfig(string View)
```

View: The name of the Web analyzer view.

Return value	Description
A <i>view_current_configuration</i> structure	The protocol server MUST return a structure that contains all the configuration options for the Web analyzer view as specified in section 2.2.1.3 .

2.2.10.6 SetConfig

This changes a global configuration value.

```
int SetConfig(string Keyword, dynamic Value)
```

Keyword: The name of a configuration option as specified in section [2.2.1.1](#).

Value: A new value for the configuration option as specified in section [2.2.1.1](#).

Return value	Description
1	The protocol server MUST return 1.

2.2.10.7 SetFDMConfig

This changes the configuration options that the protocol server use to control the analysis.

```
int SetFDMConfig(string Master, string Keyword, dynamic Value)
```

Master: A string that MUST contain the value "proc".

Keyword: The name of a configuration option as specified in section [2.2.1.2](#).

Value: A new value for the configuration option as specified in section [2.2.1.2](#).

Return value	Description
1	The protocol server MUST return 1.

2.2.10.8 SetLogLevel

This sets the log verbosity level for the protocol server. As a result of this call, the protocol server MUST alter its log verbosity level to the specified level.

```
int SetLogLevel(string Level)
```

Level: A valid log level to set. For more information, see section [2.2.7](#).

Return value	Description
1	The protocol server MUST return 1.

2.2.10.9 SetViewConfig

This changes a specific Web analyzer view configuration value.

```
int SetViewConfig(string View, string Keyword, dynamic Value)
```

View: The name of the Web analyzer view.

Keyword: The name of a configuration option as specified in section [2.2.1.3](#).

Value: A new value for the configuration option as specified in section [2.2.1.3](#).

Return value	Description
1	The protocol server MUST return 1.

2.2.11 Status Methods

Every method that is specified in this section MUST use all parameters.

2.2.11.1 GetStatus

This returns all global status information.

```
struct GetStatus()
```

Return value	Description
A <i>global_status</i> structure	The protocol server MUST return a structure that contains the global status information as specified in section 2.2.2.1 .

2.2.11.2 GetURIRelevanceData

This returns relevance information for a specific URL.

```
struct GetURIRelevanceData(string View, string ID, int AnchorCutoff)
```

View: The name of the Web analyzer view.

ID: The identifier of the document for which relevance information is being requested. This value MUST specify the identifier as a document identifier (3).

AnchorCutoff: The maximum number of entries in the *anchors* array. The entries are order by element number three, and the ones with the highest value are returned. For more information, see section [2.2.6](#) and section [2.2.6.1](#).

Return value	Description
A <i>relevance</i> structure	The protocol server returns a structure that contains URL relevance information as specified in section 2.2.6 .
An empty structure	If the Web analyzer view has not been analyzed or the document identifier could not be found, the protocol server returns an empty structure.

2.2.11.3 GetViewStatus

This returns the status of a specific Web analyzer view.

```
struct GetViewStatus(string View)
```

View: The name of the Web analyzer view.

Return value	Description
A <i>view_status</i> structure	The protocol server MUST return a structure that contains view status information as specified in section 2.2.2.2 .

2.2.12 Processing Management Methods

The processing management methods are the methods that control the order in which the Web analyzer views are processed. Every method that is specified in this section MUST use all parameters.

2.2.12.1 ForceProcessing

This forces a specific Web analyzer view to be processed as soon as possible. Any analysis that is processing MUST be paused. After the analysis of the new Web analyzer view is finished, the analysis that was paused is resumed. The new Web analyzer view, which was just processed, MUST be set to the state "scheduled", as specified in section [2.2.3.3](#).

```
int ForceProcessing(string View)
```

View: The name of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.12.2 PauseProcessing

This pauses the current analysis of a Web analyzer view. This method MUST return a fault as specified in section [2.2.8](#) if no Web analyzer view is being processed.

```
int PauseProcessing()
```

Return value	Description
1	The protocol server MUST return 1 if the analysis of a Web analyzer view was paused.
2	The protocol server MUST return 2 if the analysis is not paused yet.

2.2.12.3 ProcessOnce

This forces a specific Web analyzer view to be analyzed as soon as possible. Any analysis that is processing MUST be paused. After the analysis of the new Web analyzer view is finished, the analysis that was paused is resumed. The new Web analyzer view, which was just analyzed, MUST be set to the state of "stopped", as specified in section [2.2.3.3](#).

```
int ProcessOnce(string View)
```

View: The name of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.12.4 StartProcessing

This schedules a specific Web analyzer view for analysis. Depending on the parameters, either the Web analyzer view is moved to the first entry in the processing queue and analyzed the next time a Web analyzer view is analyzed, or its state is set to "scheduled" status and it is processed when the previously scheduled Web analyzer views have finished processing.

```
int StartProcessing(string View, int Sneak)
```

View: The name of the Web analyzer view.

Sneak: Whether to move the Web analyzer view to the front of the processing queue. This value is either 0 or 1. If 1, the Web analyzer view is scheduled to run the next time a Web analyzer view is processed. If 0, it is scheduled to run after the previously scheduled Web analyzer views.

Return value	Description
1	The protocol server MUST return 1.

2.2.12.5 StopProcessing

This sets the state of a specific Web analyzer view to "stopped". Depending on the parameters, a Web analyzer view that is currently being analyzed MUST be terminated or the process MUST finish before its status is set to "stopped", as specified in section [2.2.3.2](#).

```
int StopProcessing(string View, int Now)
```

View: The name of the Web analyzer view.

Now: Specifies whether to terminate the analysis of the Web analyzer view before setting its status to "stopped". This value is either 0 or 1. If the value is 1, any current analysis of this Web analyzer view MUST be terminated. If the value is 0, that analysis MUST finish.

Return value	Description
1	The analysis of the Web analyzer view was terminated, the Web analyzer view was not being processed, or the <i>Now</i> option was 0.
2	The protocol server MUST return 2 if the <i>Now</i> option was 1, but the analysis of the Web analyzer view could not terminate right away. This does not mean that the analysis will run until completion, only that it will be terminated later.

2.2.13 WebAnalyzer Data Management Methods

Every method that is specified in this section MUST use all parameters.

2.2.13.1 SetPreferredView

This sets the preferred Web analyzer view for a specified crawl collection.

```
int SetPreferredView(string Collection, string View)
```

Collection: The name of the crawl collection.

View: The name of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.13.2 UnsetPreferredView

This removes the preferred Web analyzer view for a specified crawl collection.

```
int UnsetPreferredView(string Collection)
```

Collection: The name of the crawl collection.

Return value	Description
1	The protocol server MUST return 1.

2.2.13.3 GetPreferredViews

This retrieves the preferred Web analyzer views for all the crawl collections.

```
struct GetPreferredViews()
```

Return value	Description
A <i>preferred_view</i> structure	The protocol server MUST return a structure that contains the preferred Web analyzer view information as specified in section 2.2.4.3 .

2.2.13.4 AddCollections

This adds one or more crawl collections to a Web analyzer view.

```
int AddCollections(array Collections, string View)
```

Collections: An array of one or more crawl collection names, where each crawl collection name is of type **string**.

View: The name of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.13.5 RemoveCollections

This removes one or more crawl collections from a Web analyzer view.

```
int RemoveCollections(array Collections, string View)
```

Collections: An array of one or more crawl collection names, where each crawl collection name is of type **string**.

View: The name of the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.13.6 AlterDescription

This alters the description of a Web analyzer view.

```
int AlterDescription(string View, string Description)
```

View: The name of the Web analyzer view.

Description: The new description for the Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

2.2.13.7 GetAllViews

This queries the protocol server for all the Web analyzer views and all associated crawl collections.

```
struct GetAllViews()
```

Return value	Description
A <i>allviews</i> structure	The protocol server MUST return a structure that contains the Web analyzer view information as specified in section 2.2.4.1 .

3 Protocol Details

3.1 Common Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

The protocol client **MUST** initiate the setup of a TCP connection between the protocol client and the protocol server. The port number for the connection **MUST** be base port plus 300.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Protocol Client Details

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The protocol client MUST call the **GetStatus** method at least once before calling any other method. If the protocol server is in Bootstrap mode as specified in section [2.2.3.2](#), or reports an error, the protocol client MUST NOT call any method that would change the state of the protocol server. In other words, the only methods that the protocol client can call are those whose names begin with "Get". This rule does not apply to the methods specified in section [3](#).

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Protocol Server Details

The protocol server listens for incoming connections, process incoming XML-RPC requests, and respond to those requests in a timely manner. If an unexpected error occurs during processing, the protocol server returns an XML-RPC fault as specified in [\[XML-RPC\]](#).

3.3.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

3.3.2 Timers

None.

3.3.3 Initialization

The initial state of the protocol server includes one Web analyzer view named "default" that MUST always be present. The protocol server has no configured crawl collections at initialization. It initiates its XML-RPC protocol server implementation as soon as it is able to process incoming requests.

The protocol server registers with the configuration component as specified in [\[MS-FSCX\]](#), and implements the following methods that are required by that protocol: **ConfigurationChanged**, **ReRegister** and **ping**. When the protocol server registers it MUST specify "WebAnalyzer" as both module type and module name, and the *alerts* array MUST contain the string "collection."

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

The following methods can alter the state of the protocol server. These altered states **MUST** always persist.

3.3.5.1 AddCollections

The successful completion of this method adds the crawl collections to the Web analyzer view. The next time the Web analyzer view is processed, the hyperlinks from the crawl collections are included in the information.

3.3.5.2 ConfigurationChanged

When this method is called and the *Alert* parameter contains the value "collection", the protocol server updates its crawl collection state. Crawl collections that are no longer valid are deleted, and new crawl collections are created.

3.3.5.3 CreateView

The successful completion of this method updates the protocol server state with the addition of the new Web analyzer view, which **MUST** be set to the state "creating", as specified in section [2.2.3.3](#). The Web analyzer view **MUST** be set to a state of "stopped" after it has been created.

3.3.5.4 DeleteCollection

The successful completion of this method initiates the deletion of all the internal state and configuration information as well as the files on disk that are specific to the crawl collection. The method **MUST** finish processing after the crawl collection is set to the state of "deleting". The crawl collection remains in this state until it has been completely removed. If a Web analyzer view that contains the crawl collection is being processed, the delete operation is postponed.

3.3.5.5 DeleteView

The successful completion of this method initiates the deletion of all the internal state and configuration information as well as the files on disk that are specific to the Web analyzer view. The method **MUST** finish processing after the Web analyzer view state has been set to the value "deleting". The Web analyzer view remains in this state until it has been completely removed. If the Web analyzer view is being processed when the method is called, the processing is terminated.

3.3.5.6 RemoveCollections

The successful completion of this method removes the crawl collections from the Web analyzer view. The next time the Web analyzer view is processed, the hyperlinks from the crawl collections **MUST NOT** be included in the information to process.

3.3.5.7 SetPreferredView

The successful completion of this method changes which Web analyzer views will be queried as specified in [\[MS-FSWASDS\]](#) when documents are processed.

3.3.5.8 UnsetPreferredView

The successful completion of this method reinitializes the mapping that specifies which Web analyzer view will be queried when documents are processed. When a crawl collection does not have a preferred Web analyzer view, the protocol server uses the default Web analyzer view.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

The examples in this section contain only the XML body for each XML-RPC message.

4.1 GetAllViews Method

In this example, the protocol client calls the **GetAllViews** method to retrieve a list of all crawl collections from which the Web analyzer views extract links. The protocol server returns two Web analyzer views, named "default" and "test". Each of these Web analyzer views contains a crawl collection named "sp".

Request:

```
<?xml version='1.0'?>
<methodCall>
  <methodName>GetAllViews</methodName>
  <params>
  </params>
</methodCall>
```

Response:

```
<?xml version='1.0'?>
<methodResponse>
  <params>
    <param>
      <value><struct>
        <member>
          <name>default</name>
          <value><struct>
            <member>
              <name>collections</name>
              <value><array><data>
                <value><string>sp</string></value>
              </data></array></value>
            </member>
            <member>
              <name>description</name>
              <value>
                <string>
                  The default WebAnalyzer view
                </string>
              </value>
            </member>
          </struct></value>
        </member>
        <member>
          <name>test</name>
          <value><struct>
            <member>
              <name>collections</name>
              <value><array><data>
                <value><string>sp</string></value>
              </data></array></value>
            </member>
            <member>
              <name>description</name>
              <value>
                <string>
                  The test WebAnalyzer view
                </string>
              </value>
            </member>
          </struct></value>
        </member>
      </struct>
    </param>
  </params>
</methodResponse>
```

```

        <name>description</name>
        <value><string></string></value>
    </member>
</struct></value>
</member>
</struct></value>
</param>
</params>
</methodResponse>

```

4.2 StartProcessing Method

Here the protocol client is requesting that the protocol server set the state of the Web analyzer view named "test" to the value "scheduled". That means that the protocol server analyzes the Web analyzer view as soon as any crawl collection in the Web analyzer view receives new information and all the other Web analyzer views in the processing queue are finished. It calls the **StartProcessing** method with "test" as argument and the "now" flag set to 0.

Request:

```

<?xml version='1.0'?>
<methodCall>
  <methodName>StartProcessing</methodName>
  <params>
    <param>
      <value><string>test</string></value>
    </param>
    <param>
      <value><int>0</int></value>
    </param>
  </params>
</methodCall>

```

Response:

```

<?xml version='1.0'?>
<methodResponse>
  <params>
    <param>
      <value><int>1</int></value>
    </param>
  </params>
</methodResponse>

```


5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

- Microsoft® FAST™ Search Server 2010

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

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

7 Change Tracking

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

8 Index

A

Abstract data model
 [client](#) 27
 [common](#) 27
 [server](#) 28
AddCollections method ([section 2.2.13.4](#) 25, [section 3.3.5.1](#) 29)
[allviews structure](#) 15
[AlterDescription method](#) 26
[Analysis stages](#) 16
[Analysis Stages message](#) 16
[Analysis substages](#) 16
[anchors array](#) 17
[Applicability](#) 8
Arrays
 [anchors](#) 17
 [processing status](#) 14
 [status](#) 13
 [system status](#) 14
 [Web analyzer view](#) 15
 [Web analyzer views](#) 15

C

[Capability negotiation](#) 8
[Change tracking](#) 35
Client
 [abstract data model](#) 27
 [higher-layer triggered events](#) 28
 [initialization](#) 27
 [message processing](#) 28
 [other local events](#) 28
 [sequencing rules](#) 28
 [timer events](#) 28
 [timers](#) 27
Configuration
 [FAST Distributed Make options](#) 10
 [global options](#) 9
 [Web analyzer view options](#) 10
[Configuration methods](#) 19
[Configuration Methods message](#) 19
[Configuration options](#) 9
[Configuration Options message](#) 9
[ConfigurationChanged method](#) 29
[Crawl collection mapping](#) 15
[Crawl collection status structure](#) 13
CreateView method ([section 2.2.9.1](#) 18, [section 3.3.5.3](#) 29)

D

Data model - abstract
 [client](#) 27
 [common](#) 27
 [server](#) 28
DeleteCollection method ([section 2.2.9.2](#) 19, [section 3.3.5.4](#) 29)

DeleteView method ([section 2.2.9.3](#) 19, [section 3.3.5.5](#) 29)

E

[Error Handling message](#) 18
Events
 [local - common](#) 27
Events - timer
 [common](#) 27
Examples
 [GetAllViews method](#) 31
 [overview](#) 31
 [StartProcessing method](#) 32

F

[FAST Distributed Make configuration options](#) 10
[Fields - vendor-extensible](#) 8
[ForceProcessing method](#) 23

G

[GetAllViews method](#) 26
[GetAllViews method example](#) 31
[GetConfig method](#) 19
[GetFDMConfig method](#) 19
[GetLogLevel method](#) 20
[GetPreferredViews method](#) 25
[GetStatus method](#) 22
[GetURIRelevanceData method](#) 22
[GetViewConfig method](#) 20
[GetViewCurrentRunConfig method](#) 20
[GetViewStatus method](#) 22
[Global configuration options](#) 9
[Global status structure](#) 11
[Glossary](#) 6

H

Higher-layer triggered events
 [client](#) 28
 [common](#) 27
 [server](#) 28

I

[Implementer - security considerations](#) 33
[Index of security parameters](#) 33
[Informative references](#) 7
Initialization
 [client](#) 27
 [common](#) 27
 [server](#) 28
[Introduction](#) 6

L

Local events

[common](#) 27
[Log Levels message](#) 17

M

Message processing

[client](#) 28
[common](#) 27
[server](#) 29

Messages

[AddCollections method](#) 25
[allViews structure](#) 15
[AlterDescription method](#) 26
[Analysis Stages](#) 16
[analysis substages](#) 16
[anchors array](#) 17
[Configuration Methods](#) 19
[Configuration Options](#) 9
[crawl collection status structure](#) 13
[CreateView method](#) 18
[DeleteCollection method](#) 19
[DeleteView method](#) 19
[Error Handling](#) 18
[ForceProcessing method](#) 23
[GetAllViews method](#) 26
[GetConfig method](#) 19
[GetFDMConfig method](#) 19
[GetLogLevel method](#) 20
[GetPreferredViews method](#) 25
[GetStatus method](#) 22
[GetURIRelevanceData method](#) 22
[GetViewConfig method](#) 20
[GetViewCurrentRunConfig method](#) 20
[GetViewStatus method](#) 22
[global status structure](#) 11
[Log Levels](#) 17
[Nested Status Arrays and Structures](#) 13
[PauseProcessing method](#) 23
[preferred Web analyzer view structure](#) 16
[Processing Management Methods](#) 23
[processing_status array](#) 14
[ProcessOnce method](#) 23
[RemoveCollections method](#) 26
[run_stats structure](#) 15
[SetConfig method](#) 20
[SetFDMConfig method](#) 21
[SetLogLevel method](#) 21
[SetPreferredView method](#) 25
[SetViewConfig method](#) 21
[StartProcessing method](#) 24
[Status Methods](#) 22
[Status Structures](#) 11
[StopProcessing method](#) 24
[syntax](#) 9
[system_status array](#) 14
[transport](#) 9
[UnsetPreferredView method](#) 25
[URL Relevance Structure](#) 17
[view status structure](#) 12
[Web Analyzer View and Crawl Collection Mappings](#) 15
[Web analyzer views array](#) 15

[WebAnalyzer Data Management Methods](#) 24
[WebAnalyzer Management Methods](#) 18

Methods

[AddCollections](#) ([section 2.2.13.4](#) 25, [section 3.3.5.1](#) 29)
[AlterDescription](#) 26
[configuration](#) 19
[ConfigurationChanged](#) 29
[CreateView](#) ([section 2.2.9.1](#) 18, [section 3.3.5.3](#) 29)
[DeleteCollection](#) ([section 2.2.9.2](#) 19, [section 3.3.5.4](#) 29)
[DeleteView](#) ([section 2.2.9.3](#) 19, [section 3.3.5.5](#) 29)
[ForceProcessing](#) 23
[GetAllViews](#) 26
[GetConfig](#) 19
[GetFDMConfig](#) 19
[GetLogLevel](#) 20
[GetPreferredViews](#) 25
[GetStatus method](#) 22
[GetURIRelevanceData](#) 22
[GetViewConfig](#) 20
[GetViewCurrentRunConfig](#) 20
[GetViewStatus](#) 22
[PauseProcessing](#) 23
[processing management](#) 23
[ProcessOnce](#) 23
[RemoveCollections](#) ([section 2.2.13.5](#) 26, [section 3.3.5.6](#) 29)
[SetConfig](#) 20
[SetFDMConfig](#) 21
[SetLogLevel](#) 21
[SetPreferredView](#) ([section 2.2.13.1](#) 25, [section 3.3.5.7](#) 29)
[SetViewConfig](#) 21
[StartProcessing](#) 24
[status](#) 22
[StopProcessing](#) 24
[UnsetPreferredView](#) ([section 2.2.13.2](#) 25, [section 3.3.5.8](#) 30)
[WebAnalyzer data management](#) 24
[WebAnalyzer management](#) 18

N

[Nested status arrays and structures](#) 13
[Nested Status Arrays and Structures message](#) 13
[Normative references](#) 6

O

Options

[configuration](#) 9
Other local events
[client](#) 28
[server](#) 30
[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 33

[PauseProcessing method](#) 23
[Preconditions](#) 8
[Preferred Web analyzer view structure](#) 16
[Prerequisites](#) 8
[Processing management methods](#) 23
[Processing Management Methods message](#) 23
[processing_status array](#) 14
[ProcessOnce method](#) 23
[Product behavior](#) 34

R

References
 [informative](#) 7
 [normative](#) 6
[Relationship to other protocols](#) 7
RemoveCollections method ([section 2.2.13.5](#) 26,
 [section 3.3.5.6](#) 29)
[run_stats structure](#) 15

S

Security
 [implementer considerations](#) 33
 [parameter index](#) 33
Sequencing rules
 [client](#) 28
 [common](#) 27
 [server](#) 29
Server
 [abstract data model](#) 28
 [higher-layer triggered events](#) 28
 [initialization](#) 28
 [message processing](#) 29
 [other local events](#) 30
 [overview](#) 28
 [sequencing rules](#) 29
 [timer events](#) 30
 [timers](#) 28
 [SetConfig method](#) 20
 [SetFDMConfig method](#) 21
 [SetLogLevel method](#) 21
 SetPreferredView method ([section 2.2.13.1](#) 25,
 [section 3.3.5.7](#) 29)
 [SetViewConfig method](#) 21
 [Standards assignments](#) 8
 [StartProcessing method](#) 24
 [StartProcessing method example](#) 32
 [Status methods](#) 22
 [Status Methods message](#) 22
 [Status structures](#) 11
 [Status Structures message](#) 11
 [StopProcessing method](#) 24
Structure
 [URL relevance](#) 17
Structures
 [allviews](#) 15
 [crawl collection status](#) 13
 [global status](#) 11
 [preferred Web analyzer view](#) 16
 [run_stats](#) 15
 status ([section 2.2.2](#) 11, [section 2.2.3](#) 13)

[view status](#) 12
 [Web analyzer view](#) 15
Syntax
 [messages - overview](#) 9
 [system_status array](#) 14

T

Timer events
 [client](#) 28
 [common](#) 27
 [server](#) 30
Timers
 [client](#) 27
 [common](#) 27
 [server](#) 28
[Tracking changes](#) 35
[Transport](#) 9
Triggered events – higher layer
 [common](#) 27
Triggered events - higher-layer
 [client](#) 28
 [server](#) 28

U

UnsetPreferredView method ([section 2.2.13.2](#) 25,
 [section 3.3.5.8](#) 30)
[URL relevance structure](#) 17
[URL Relevance Structure message](#) 17

V

[Vendor-extensible fields](#) 8
[Versioning](#) 8
[View status structure](#) 12

W

[Web analyzer view](#) 15
[Web Analyzer View and Crawl Collection Mappings message](#) 15
[Web analyzer view configuration options](#) 10
[Web analyzer views array](#) 15
[WebAnalyzer data management methods](#) 24
[WebAnalyzer Data Management Methods message](#) 24
[WebAnalyzer management methods](#) 18
[WebAnalyzer Management Methods message](#) 18