

# [MS-FSWASMT]: WebAnalyzer and SPRel Multinode Transport 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](mailto: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
02/19/2010	1.0	Major	Initial Availability
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	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.

# Table of Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	6
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
<b>2 Messages</b>	<b>8</b>
2.1 Transport	8
2.2 Common Data Types	8
2.2.1 Updated Crawl Collections	8
2.2.1.1 Crawl Collection Status	8
2.2.1.2 Table Counters	8
2.2.2 Error Handling	9
<b>3 Protocol Details</b>	<b>10</b>
3.1 Lookup Database Component Details	10
3.1.1 Abstract Data Model	10
3.1.2 Timers	10
3.1.3 Initialization	10
3.1.4 Message Processing Events and Sequencing Rules	10
3.1.4.1 add_view	10
3.1.4.2 delete_db	11
3.1.4.3 get_attributes	12
3.1.4.4 remove_view	12
3.1.4.5 switch_db	12
3.1.5 Timer Events	13
3.1.6 Other Local Events	13
3.2 Link Processing Component Details	13
3.2.1 Abstract Data Model	13
3.2.2 Timers	13
3.2.3 Initialization	13
3.2.4 Message Processing Events and Sequencing Rules	14
3.2.4.1 DeleteCollection	14
3.2.4.2 GetUpdatedCollections	14
3.2.4.3 RotateLogs	15
3.2.5 Timer Events	15
3.2.6 Other Local Events	15
<b>4 Protocol Examples</b>	<b>16</b>
4.1 GetUpdatedCollections	16
4.2 RotateLogs	16
4.3 switch_db	17
4.4 get_attributes	18

<b>5 Security</b> .....	<b>20</b>
5.1 Security Considerations for Implementers.....	20
5.2 Index of Security Parameters .....	20
<b>6 Appendix A: XML Schema</b> .....	<b>21</b>
<b>7 Appendix B: Product Behavior</b> .....	<b>24</b>
<b>8 Change Tracking</b> .....	<b>25</b>
<b>9 Index</b> .....	<b>26</b>

# 1 Introduction

This document specifies the WebAnalyzer and SPRel Multinode Transport Protocol, which manages the messages exchanged between hosts in a distributed system when a central component communicates with other components.

## 1.1 Glossary

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

### **Augmented Backus-Naur Form (ABNF)**

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

**associated query**  
**crawl collection**  
**document identifier**  
**hyperlink**  
**search clickthrough**  
**Web analyzer view**

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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-FSCX] Microsoft Corporation, "[Configuration \(XML-RPC\) Protocol Specification](#)", November 2009.

[MS-FSFDMW] Microsoft Corporation, "[FAST Distributed Make Worker Protocol Specification](#)", February 2010.

[MS-FSSPRADM] Microsoft Corporation, "[SPRel Administration and Status Protocol Specification](#)", November 2009.

[MS-FSSPRDF] Microsoft Corporation, "[SPRel Data File Format](#)", February 2010.

[MS-FSWAADM] Microsoft Corporation, "[WebAnalyzer Administration and Status Protocol Specification](#)", November 2009.

[MS-FSWADF] Microsoft Corporation, "[WebAnalyzer Data File Format](#)", February 2010.

[MS-FSWASDR] Microsoft Corporation, "[WebAnalyzer/SPRel Data Receiving Protocol Specification](#)", November 2009.

[MS-FSWASDS] Microsoft Corporation, "[WebAnalyzer/SPRel Data Serving Protocol Specification](#)", November 2009.

[MS-FSXTAPI] Microsoft Corporation, "[XML-RPC Translatable API Structure Specification](#)", February 2010.

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

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

### 1.2.2 Informative References

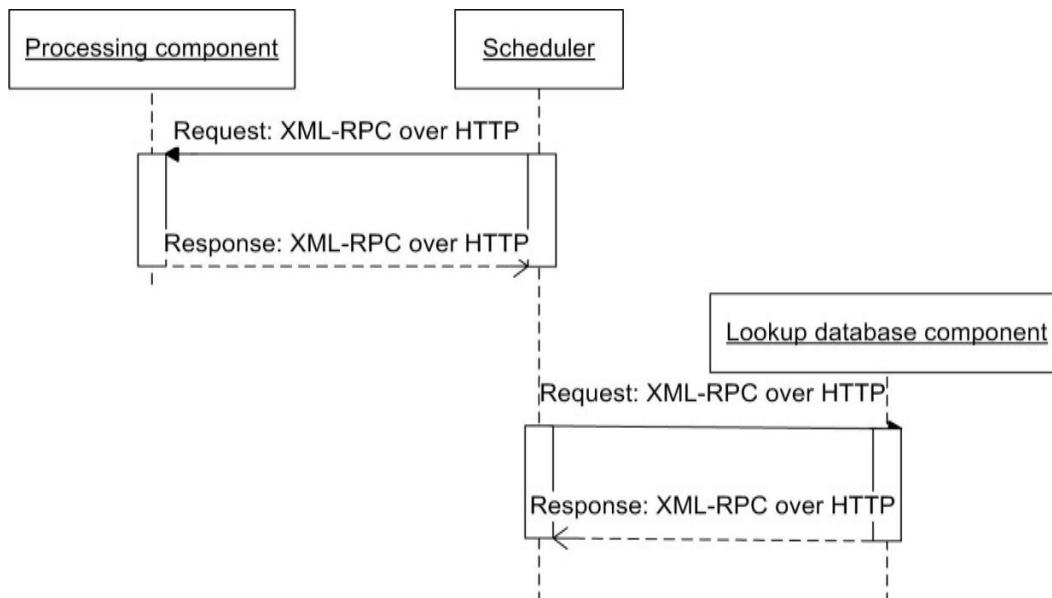
[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)", June 2008.

### 1.3 Protocol Overview (Synopsis)

This protocol specifies how a protocol client issues requests to other components in a **hyperlink** or **search clickthrough** analysis system. This protocol allows such a system to be distributed over multiple protocol servers or virtual servers, each of which performs a subset of the analysis. The system can then scale CPU and I/O resources by balancing the load between servers.

A hyperlink or search clickthrough analysis system consists of three services, as shown in the following figure.

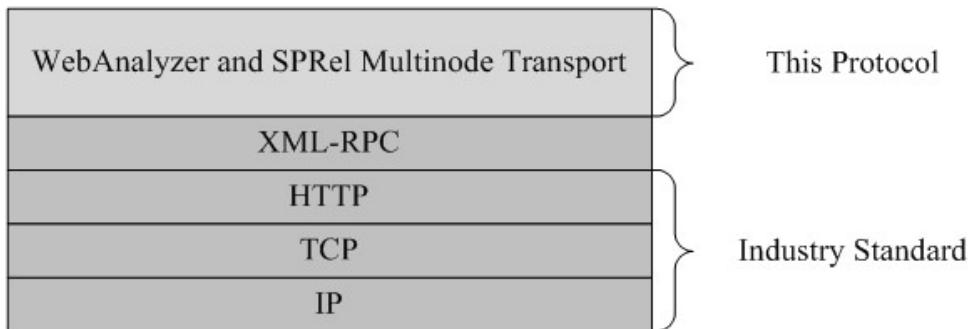


**Figure 1: Protocol communication overview**

The scheduler is the protocol client that orchestrates the analysis. The link processing component performs the analysis, and the lookup database component transmits the output. There can be multiple link processing components and lookup database components in the system.

## 1.4 Relationship to Other Protocols

The protocol uses XML-RPC over HTTP as shown in the following diagram:



**Figure 2: This protocol in relation to other protocols**

## 1.5 Prerequisites/Preconditions

The protocol client obtains the host name and the port for the protocol server before initiating the protocol.

The lookup database component listens on port number base port plus 295, and the link processing component listens on port number base port plus 290.

## 1.6 Applicability Statement

This protocol is used in a distributed system in which a central scheduling component communicates with other components in a hyperlink or search clickthrough analysis system.

## 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 Common Data Types

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**
- **int**
- **string**
- **struct**

#### 2.2.1 Updated Crawl Collections

The following table specifies the content of the structure that the **GetUpdatedCollections** method returns.

Member name	Type	Description
<b>crawl collection</b> name	<b>array</b>	An array that contains crawl collection status information as specified in section <a href="#">2.2.1.1</a> .

##### 2.2.1.1 Crawl Collection Status

The **array** that is part of the structure specified in section [2.2.1](#) MUST be as specified in the following table.

Element	Type	Description
0	<b>struct</b>	A <b>struct</b> that contains table counters as specified in section <a href="#">2.2.1.2</a> .
1	<b>int</b>	Specifies the number of seconds elapsed since the last time the protocol server received information for the crawl collection.

##### 2.2.1.2 Table Counters

The content of the structure that the protocol server returns to the protocol client MUST be as specified in the following table. The members in this structure match the **key** entries specified in [\[MS-FSWASDR\]](#) section 2.2.1.3.1.1 through [2.2.1.3.1.7](#) if the **application** entry contains the value "webanalyzer". If the **application** entry contains the value "sharepoint", then the members in this structure match the **key** entries specified in section [\[MS-FSWASDR\]](#) section 2.2.1.3.2.1.



Member name	Type	Description
table name	<b>int</b>	Specifies the number of records that the protocol server received for the table.

## 2.2.2 Error Handling

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

Most errors that occur generate fault messages. Whenever a method generates a fault message, it substitutes the fault message 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 **MUST** be of type **int** and contain the value 1. The fault **string** **MUST** be as specified in the following **Augmented Backus-Naur Form (ABNF)** rules:

```

errormsg = (e_prefix stderror / a_prefix attrerror) delim errortxt

delim = %d39.38.103.116.59.58.32
a_prefix = %d38.108.116.59.99.108.97.115.115.32.39
e_prefix = %d38.108.116.59.116.121.112.101.32.39

stderror = %d101.120.99.101.112.116.105.111.110.46 (exception / attributeerror)
attrerror = %d97.116.116.114.105.98.117.116.101.115.116.111.114.97.103.101.46 (notfound /
emptytable)

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
notfound = %d78.111.116.70.111.117.110.100
emptytable = %d69.109.112.116.121.84.97.98.108.101

errortxt = 1*(VCHAR / SP)

```

**exception:** The method or called application returns this fault message when the fault occurred within the method.

**attributeerror:** The method or called application returns this fault message when it calls an unknown method.

**notfound:** The **get\_attributes** method returns this fault message if the *Key* parameter is not found.

**emptytable:** The **get\_attributes** method returns this fault message if the database is empty.

## 3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

### 3.1 Lookup Database Component Details

The following sections specify the details of the lookup database component, which transmits the output of analysis performed by the link processing component specified in section [3.2](#).

#### 3.1.1 Abstract Data Model

None.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

If the lookup database component runs on a protocol server or virtual server that does not have a link processing component, the lookup database component registers with the configuration component as specified in [\[MS-FSCX\]](#), and implements the **ConfigurationChanged**, **ReRegister** and **ping** methods as required by that protocol. When the protocol server registers it MUST specify "fdmworker" as both module type and module name.

#### 3.1.4 Message Processing Events and Sequencing Rules

The following table lists the methods that the lookup database component MUST support as specified in the following sections.

Name	Description
add_view	Add a new <b>Web analyzer view</b> to the lookup database component as specified in section <a href="#">3.1.4.1</a> .
delete_db	Delete database files for a given Web analyzer view as specified in section <a href="#">3.1.4.2</a> .
get_attributes	Retrieves relevant results, a search result that is relevant to a query term based on rank, for a specified <b>document identifier (3)</b> as specified in section <a href="#">3.1.4.3</a> . By default, the higher the rank, the higher the item appears in the query results.
remove_view	This deletes an existing Web analyzer view as specified in section <a href="#">3.1.4.4</a> .
switch_db	Replaces the current database files that were used for a specific view with another set of files as specified in section <a href="#">3.1.4.5</a> .

The following sections specify how the protocol methods alter the state of the protocol server.

##### 3.1.4.1 add\_view

Add a new Web analyzer view to the lookup database component.

```
int add_view(string View, string Binfilename, string Idxfilename)
```

**View:** The name of the Web analyzer view. In a search clickthrough analysis system this field MUST contain the value "sharepoint\_rel", and in a hyperlink analysis system this MUST contain the name of a valid Web analyzer view.

**Binfilename:** In a search clickthrough analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSSPRDF\]](#) section 2.5.1. In a hyperlink analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSWADF\]](#) section 2.5.1.

**Idxfilename:** In a search clickthrough analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSSPRDF\]](#) section 2.5.2. The directory MUST also contain a file which name begins with the same file name, but which suffix MUST contain the value ".idx.ofs", as specified in [\[MS-FSSPRDF\]](#) section 2.5.3. In a hyperlink analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSWADF\]](#) section 2.5.2. The directory MUST also contain a file which name begins with the same file name, but which suffix MUST contain the value ".idx.ofs", as specified in [\[MS-FSWADF\]](#) [\[MS-FSWADF\]](#) section 2.5.3.

Return value	Description
1	The protocol server MUST return 1.

When the protocol server processes this method, it MUST add the new Web analyzer view to its internal state and send data from the databases in this Web analyzer view.

### 3.1.4.2 delete\_db

This method deletes database files from disk for a given Web analyzer view. It is called only after **remove\_view** or **switch\_db** is called for the Web analyzer view. The Web analyzer view MUST first be removed with the **remove\_view** method.

```
int delete_db(string View, string Binfilename, string Idxfilename)
```

**View:** The name of the Web analyzer view. In a search clickthrough analysis system this field MUST contain the value "sharepoint\_rel", and in a hyperlink analysis system this MUST contain the name of a valid Web analyzer view.

**Binfilename:** In a search clickthrough analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSSPRDF\]](#) section 2.5.1. In a hyperlink analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSWADF\]](#) section 2.5.1.

**Idxfilename:** In a search clickthrough analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSSPRDF\]](#) section 2.5.2. The directory MUST also contain a file which name begins with the same file name, but which suffix MUST contain the value ".idx.ofs", as specified in [\[MS-FSSPRDF\]](#) section 2.5.3. In a hyperlink analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSWADF\]](#) section 2.5.2. The directory MUST also contain a file which name begins with the same file name, but which suffix MUST contain the value ".idx.ofs", as specified in [\[MS-FSWADF\]](#) section 2.5.3.

Return value	Description
1	The protocol server MUST return 1.

### 3.1.4.3 get\_attributes

This retrieves a structure that contains relevance information for a specified document identifier (3). In installations that use multiple lookup database components, the protocol client MUST use the Key parameter to find the protocol server to query, as specified in [\[MS-FSWASDS\]](#) section 3.1.3.5.

```
struct get_attributes(string View, string Key)
```

**View:** The name of the Web analyzer view. In a search clickthrough analysis system this field MUST contain the value "sharepoint\_rel", and in a hyperlink analysis system this MUST contain the name of a valid Web analyzer view.

**Key:** The identifier of the document for which the protocol client requests relevance information. This field MUST be specified as a document identifier (3).

Return value	Description
struct	In a hyperlink analysis system the protocol server MUST return a structure that contains relevance information as specified in <a href="#">[MS-FSWAADM]</a> section 2.2.6. In a search clickthrough analysis system the protocol server MUST return a structure that contains relevance information as specified in <a href="#">[MS-FSSPRADM]</a> section 2.2.6.
fault	If the specified Key parameter is not known to the protocol server it MUST return a <b>notfound</b> fault as specified in section <a href="#">2.2.2</a> .

### 3.1.4.4 remove\_view

This deletes an existing Web analyzer view.

```
int remove_view(string View)
```

**View:** The name of the Web analyzer view. In a search clickthrough analysis system this field MUST contain the value "sharepoint\_rel", and in a hyperlink analysis system this MUST contain the name of a valid Web analyzer view.

Return value	Description
1	The protocol server MUST return 1.

This method means that the protocol server will no longer send data for the Web analyzer view.

### 3.1.4.5 switch\_db

This method replaces the current database files that were used for a specific view with another set of files.

```
int switch_db(string View, string Binfilename, string Idxfilename)
```

**View:** The name of the Web analyzer view. In a search clickthrough analysis system this field MUST contain the value "sharepoint\_rel", and in a hyperlink analysis system this MUST contain the name of a valid Web analyzer view.

**Binfilename:** In a search clickthrough analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSSPRDF\]](#) section 2.5.1. In a hyperlink analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSWADF\]](#) section 2.5.1.

**Idxfilename:** In a search clickthrough analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSSPRDF\]](#) section 2.5.2. The directory MUST also contain a file which name begins with the same file name, but which suffix MUST contain the value ".idx ofs", as specified in [\[MS-FSSPRDF\]](#) section 2.5.3. In a hyperlink analysis system this field MUST contain the name of the path that points to a file as specified in [\[MS-FSWADF\]](#) section 2.5.2. The directory MUST also contain a file which name begins with the same file name, but which suffix MUST contain the value ".idx ofs", as specified in [\[MS-FSWADF\]](#) section 2.5.3.

Return value	Description
1	The protocol server MUST return 1.

This method MUST update the state for the Web analyzer view so that it sends relevance information from the new database files. The **delete\_db** method deletes the database files that were in use.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.

## 3.2 Link Processing Component Details

### 3.2.1 Abstract Data Model

None.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

The link processing component MUST register two entries with the configuration component as specified in [\[MS-FSCX\]](#), and implements the **ConfigurationChanged**, **ReRegister** and **ping** methods as required by that protocol. When the protocol registers the first entry, the module type and module name parameters both contain the value "WaLinkStorerReceiver". The second entry module type and module name parameters both contain the value "fdmworker".

### 3.2.4 Message Processing Events and Sequencing Rules

The following table lists the methods used by the link processing component, as specified in the following sections.

Name	Description
DeleteCollection	Deletes a crawl collection as specified in section <a href="#">3.2.4.1</a> .
GetUpdatedCollections	Returns a structure that contains all updated crawl collections as specified in section <a href="#">3.2.4.2</a> .
RotateLogs	Prepares to analyze the information received by the protocol client as specified in section <a href="#">3.2.4.3</a> .

The following methods alter the state of the protocol server. These altered states MUST be persisted in stable storage.

#### 3.2.4.1 DeleteCollection

This method specifies that the protocol server removes a crawl collection. It initiates the deletion of all the internal state for the crawl collection. Information is deleted as specified in [\[MS-FSFDMW\]](#).

```
int DeleteCollection(string Application, string Collection)
```

**Application:** A **string** that MUST contain the value "webanalyzer" for a hyperlink analysis system or "sharepoint" for a search clickthrough analysis system.

**Collection:** The name of the crawl collection.

Return value	Description
1	The protocol server MUST return 1.

#### 3.2.4.2 GetUpdatedCollections

This returns a structure that contains all crawl collections that have been updated since the last time the RotateLogs method was called.

```
struct GetUpdatedCollections(string Application)
```

**Application:** A **string** that MUST contain the value "webanalyzer" for a hyperlink analysis system or "sharepoint" for a search clickthrough analysis system.

Type	Description
Structure	This structure MUST contain information about updated crawl collections as specified in section <a href="#">2.2.1</a> . If no updated collections exist, the protocol server returns an empty structure.

### 3.2.4.3 RotateLogs

This specifies that the protocol server prepares to analyze the information it received, as specified in [\[MS-FSWASDR\]](#). The processes the protocol server performs to prepare for the analysis are implementation dependent.

```
int RotateLogs(string Application, array Collections)
```

**Application:** A **string** that MUST contain the value "webanalyzer" for a hyperlink analysis system or "sharepoint" for a search clickthrough analysis system.

**Collections:** The name of the crawl collections to prepare.

Return value	Description
0	The protocol server MUST return 0 if none of the crawl collections has received new information.
1	The protocol server MUST return 1 if any of the crawl collections has received new documents since the last time <b>RotateLogs</b> was called.

When this method is executed the protocol server MUST prepare all received information for analysis, and reset all counters returned by the **GetUpdatedCollections** method.

### 3.2.5 Timer Events

None.

### 3.2.6 Other Local Events

None.

## 4 Protocol Examples

The following sections demonstrate a scenario where the scheduler component in a search clickthrough analysis system communicates with the link processing component before beginning an analysis run, and communicates with the lookup database component after the analysis finishes.

### 4.1 GetUpdatedCollections

The scheduler component requests information about new documents from the link processing component. The link processing component responds that 17 records were received and that the most recent record was received 44 seconds ago.

#### Request:

```
<?xml version='1.0'?>
<methodCall>
  <methodName>GetUpdatedCollections</methodName>
  <params>
    <param>
      <value><string>sharepoint</string></value>
    </param>
  </params>
</methodCall>
```

#### Response:

```
<?xml version='1.0'?>
<methodResponse>
  <params>
    <param>
      <value><struct>
        <member>
          <name>sp</name>
          <value><array><data>
            <value><struct>
              <member>
                <name>uris</name>
                <value><int>17</int></value>
              </member>
            </struct></value>
          <value><int>44</int></value>
        </data></array></value>
      </member>
    </struct></value>
  </param>
</params>
</methodResponse>
```

### 4.2 RotateLogs

The scheduler component begins an analysis run. It issues a RotateLogs command to the link processing component. The link processing component responds with the value 1, because it updated its crawl collections with new documents.

#### Request:



```

<?xml version='1.0'?>
<methodCall>
  <methodName>RotateLogs</methodName>
  <params>
    <param>
      <value><string>sharepoint</string></value>
    </param>
    <param>
      <value><array><data>
        <value><string>sp</string></value>
      </data></array></value>
    </param>
  </params>
</methodCall>

```

**Response:**

```

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

```

### 4.3 switch\_db

When the link processing component produces a new database, the schedule component sends a request to the lookup database component to switch to the new database files.

**Request:**

```

<?xml version='1.0'?>
<methodCall>
  <methodName>switch_db</methodName>
  <params>
    <param>
      <value><string>sharepoint_rel</string></value>
    </param>
    <param>
      <value><string>C:\fastsearch\data\webanalyzer\walookupdb\5.sharepoint_rel.0.0.bin</string></value>
    </param>
    <param>
      <value><string>C:\fastsearch\data\webanalyzer\walookupdb\5.sharepoint_rel.0.0.idx</string></value>
    </param>
  </params>
</methodCall>

```

**Response:**

```

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

```

#### 4.4 get\_attributes

After the analysis finishes, the scheduler component queries the lookup database component for relevance information about a document which document identifier (3) contains the value "ssic://888104193". The lookup database component sends a struct that shows that this document contains two **associated queries**.

##### Request:

```

<?xml version='1.0'?>
<methodCall>
  <methodName>get_attributes</methodName>
  <params>
    <param>
      <value><string>sharepoint_rel</string></value>
    </param>
    <param>
      <value><string>ssic://888104193</string></value>
    </param>
  </params>
</methodCall>

```

##### Response:

```

<?xml version='1.0'?>
<methodResponse>
  <params>
    <param>
      <value><struct>
        <member>
          <name>contentid</name>
          <value><string>ssic://888104193</string></value>
        </member>
        <member>
          <name>queries</name>
          <value><array><data>
            <value><array><data>
              <value><string>4</string></value>
              <value><string>6</string></value>
              <value><string>1</string></value>
              <value><string>2</string></value>
              <value><string>contoso pharmaceuticals</string></value>
            </data></array></value>
          <value><array><data>
            <value><string>2</string></value>
            <value><string>2</string></value>
            <value><string>1</string></value>
          </data></array></value>
        </member>
      </struct>
    </param>
  </params>

```

```
        <value><string>1</string></value>
        <value><string>contoso</string></value>
    </data></array></value>
</data></array></value>
</member>
</struct></value>
</param>
</params>
</methodResponse>
```

## **5 Security**

### **5.1 Security Considerations for Implementers**

None.

### **5.2 Index of Security Parameters**

None.

## 6 Appendix A: XML Schema

For ease of implementation the following XML-RPC Schema is provided.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="methodCall">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name="methodName">
          <xsd:simpleType>
            <xsd:restriction base="ASCIIString">
              <xsd:pattern value="([A-Za-z0-9]||\.|:|_|)"* />
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="params" minOccurs="0" maxOccurs="1">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="param" type="ParamType"
                minOccurs="0" maxOccurs="unbounded" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="methodResponse">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element name="params">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="param" type="ParamType" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="fault">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="struct">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="member"
                            type="MemberType" />
                          <xsd:element name="member"
                            type="MemberType" />
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>

<xsd:complexType name="ParamType">
    <xsd:sequence>
        <xsd:element name="value" type="ValueType" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ValueType" mixed="true">
    <xsd:choice>
        <xsd:element name="i4" type="xsd:int" />
        <xsd:element name="int" type="xsd:int" />
        <xsd:element name="string" type="ASCIIString" />
        <xsd:element name="double" type="xsd:decimal" />
        <xsd:element name="Base64" type="xsd:base64Binary" />
        <xsd:element name="boolean" type="NumericBoolean" />
        <xsd:element name="dateTime.iso8601" type="xsd:dateTime" />
        <xsd:element name="array" type="ArrayType" />
        <xsd:element name="struct" type="StructType" />
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="StructType">
    <xsd:sequence>
        <xsd:element name="member" type="MemberType"
            maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MemberType">
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string" />
        <xsd:element name="value" type="ValueType" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ArrayType">
    <xsd:sequence>
        <xsd:element name="data">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="value" type="ValueType"
                        minOccurs="0" maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="ASCIIString">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="([ -~]|\n|\r|\t)*" />
    </xsd:restriction>
</xsd:simpleType>

```

```
<xsd:simpleType name="NumericBoolean">  
  <xsd:restriction base="xsd:boolean">  
    <xsd:pattern value="0|1" />  
  </xsd:restriction>  
</xsd:simpleType>  
</xsd:schema>
```

## 7 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.



## 8 Change Tracking

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

## 9 Index

### A

Abstract data model  
[link processing component](#) 13  
[lookup database component](#) 10  
[Applicability](#) 7

### C

[Capability negotiation](#) 7  
[Change tracking](#) 25  
[Common data types](#) 8

### D

Data model - abstract  
[link processing component](#) 13  
[lookup database component](#) 10  
Data types  
[common - overview](#) 8  
[crawl collection status](#) 8  
[table counters](#) 8  
[updated crawl collections](#) 8

### E

[Error handling - messages](#) 9  
Events - local  
[link processing component](#) 15  
[lookup database component](#) 13  
Examples  
[get\\_attributes method](#) 18  
[GetUpdatedCollections method](#) 16  
[overview](#) 16  
[RotateLogs method](#) 16  
[switch\\_db method](#) 17

### F

[Fields - vendor-extensible](#) 7

### G

[get\\_attributes method example](#) 18  
[GetUpdatedCollections method example](#) 16  
[Glossary](#) 5

### I

[Implementer - security considerations](#) 20  
[Index of security parameters](#) 20  
[Informative references](#) 6  
Initialization  
[link processing component](#) 13  
[lookup database component](#) 10  
[Introduction](#) 5

### L

Link processing component  
[abstract\\_data\\_model](#) 13  
[DeleteCollection method](#) 14  
[GetUpdatedCollections method](#) 14  
[initialization](#) 13  
[local events](#) 15  
[message processing](#) 14  
[RotateLogs method](#) 15  
[sequencing rules](#) 14  
[timer events](#) 15

#### Local events

[link processing component](#) 15  
[lookup database component](#) 13

#### Lookup database component

[abstract\\_data\\_model](#) 10  
[add\\_view method](#) 10  
[delete\\_db method](#) 11  
[get\\_attributes method](#) 12  
[initialization](#) 10  
[local events](#) 13  
[message processing](#) 10  
[overview](#) 10  
[remove\\_view method](#) 12  
[sequencing rules](#) 10  
[switch\\_db method](#) 12  
[timer events](#) 13  
[timers](#) 10

### M

Message processing  
[add\\_view method](#) 10  
[delete\\_db method](#) 11  
[get\\_attributes method](#) 12  
[GetUpdatedCollections method](#) 14  
[link processing component](#) 14  
[lookup database component](#) 10  
[remove\\_view method](#) 12  
[switch\\_db method](#) 12

#### Messages

[common data types](#) 8  
[crawl collection status data type](#) 8  
[error handling](#) 9  
[table counters data type](#) 8  
[transport](#) 8  
[updated crawl collections data type](#) 8

#### Methods

[add\\_view](#) 10  
[delete\\_db](#) 11  
[DeleteCollection](#) 14  
[get\\_attributes](#) 12  
[GetUpdatedCollections](#) 14  
[remove\\_view](#) 12  
[RotateLogs](#) 15  
[switch\\_db](#) 12

### N

[Normative references](#) 5

## O

[XML Schema](#) 21

[Overview \(synopsis\)](#) 6

## P

[Parameters - security index](#) 20

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 24

## R

References

[informative](#) 6

[normative](#) 5

[Relationship to other protocols](#) 7

[RotateLogs method example](#) 16

## S

[Schema - XML](#) 21

Security

[implementer considerations](#) 20

[parameter index](#) 20

Sequencing rules

[add\\_view method](#) 10

[delete\\_db method](#) 11

[DeleteCollection method](#) 14

[get\\_attributes method](#) 12

[GetUpdatedCollections method](#) 14

[link\\_processing component](#) 14

[lookup database component](#) 10

[remove\\_view method](#) 12

[RotateLogs method](#) 15

[switch\\_db method](#) 12

[Standards assignments](#) 7

[switch\\_db method example](#) 17

## T

Timer events

[link\\_processing component](#) 15

[lookup database component](#) 13

Timers

[link\\_processing component](#) 13

[lookup database component](#) 10

[Tracking changes](#) 25

[Transport](#) 8

## U

[Updated crawl collections data type](#) 8

[crawl collection status data type](#) 8

[table counters data type](#) 8

## V

[Vendor-extensible fields](#) 7

[Versioning](#) 7

## X