

[MS-CIPROP]: Index Propagation 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 [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **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
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Changes made for template compliance
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.5	Minor	Clarified the meaning of the technical content.
04/11/2012	2.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.5	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	7
1.2.2 Informative References	7
1.3 Protocol Overview (Synopsis)	7
1.4 Relationship to Other Protocols	9
1.5 Prerequisites/Preconditions	10
1.6 Applicability Statement	10
1.7 Versioning and Capability Negotiation	10
1.8 Vendor-Extensible Fields	10
1.9 Standards Assignments	10
2 Messages	11
2.1 Transport	11
2.2 Message Syntax	11
2.2.1 Common Data Types	11
2.2.1.1 Simple Data Types and Enumerations	11
2.2.1.2 Enumerations	11
2.2.1.2.1 Task Type	11
2.2.1.2.2 CatalogID	11
2.2.2 Full-Text Index Component Message	11
2.2.2.1 Propagation List File	12
2.2.2.1.1 String Record	12
2.2.2.1.2 String List	12
2.2.2.3 Versioned Index Identifier	12
3 Protocol Details	14
3.1 Back-End Database Server Details	14
3.1.1 Abstract Data Model	14
3.1.1.1 List of Ready Query Components	14
3.1.1.2 List of Running Tasks	14
3.1.2 Timers	15
3.1.3 Initialization	15
3.1.4 Higher-Layer Triggered Events	15
3.1.5 Message Processing Events and Sequencing Rules	15
3.1.5.1 proc_MSS_PropagationIndexerCleanUpTablesForTask	16
3.1.5.2 proc_MSS_PropagationIndexerGetCompletedTasks	16
3.1.5.2.1 Completed Tasks Result Set	17
3.1.5.3 proc_MSS_PropagationIndexerGetReadyQueryComponents	17
3.1.5.3.1 Ready Query Components Result Set	18
3.1.5.4 proc_MSS_PropagationIndexerGetTasks	18
3.1.5.4.1 Propagation Tasks Result Set	18
3.1.5.5 proc_MSS_PropagationIndexerInsertNewTask	19
3.1.5.6 proc_MSS_PropagationQueryComponentPickUpNewPropagationItems	20
3.1.5.6.1 Propagation Tasks Result Set	21
3.1.5.7 proc_MSS_PropagationIndexerDeleteAllTasksFromSender	22
3.1.5.8 proc_MSS_PropagationIndexerGetActiveIndexPartitionGuids	22
3.1.5.8.1 Partitions Result Set	22
3.1.5.9 proc_MSS_PropagationIndexerGetActiveIndexPartitionHashes	23

3.1.5.9.1	Partition Hashes.....	23
3.1.5.10	proc_MSS_PropagationQueryComponentReportTaskReady	23
3.1.6	Timer Events	24
3.1.7	Other Local Events	24
3.2	Sender Details.....	24
3.2.1	Abstract Data Model	24
3.2.1.1	Search Application Name	25
3.2.1.2	Sender Identifier	25
3.2.1.3	List of Ready Query Components	25
3.2.1.4	List of Completed Tasks.....	25
3.2.2	Timers	25
3.2.3	Initialization	25
3.2.4	Higher-Layer Triggered Events.....	25
3.2.5	Message Processing Events and Sequencing Rules.....	25
3.2.5.1	Sending a proc_MSS_PropagationIndexerGetReadyQueryComponents Message	25
3.2.5.2	Receiving a Ready Query Components Result Set	26
3.2.5.3	Sending a Full-Text Index Component Message	26
3.2.5.4	Sending a proc_MSS_PropagationIndexerInsertNewTask Message	27
3.2.5.5	Sending a proc_MSS_PropagationIndexerGetCompletedTasks Message	28
3.2.5.6	Receiving a Completed Tasks Result Set Message	28
3.2.5.7	Sending a proc_MSS_PropagationIndexerCleanUpTablesForTask Message	28
3.2.5.8	Sending a proc_MSS_PropagationIndexerGetTasks Message	29
3.2.5.9	Receiving a Propagation Tasks Result Set	29
3.2.6	Timer Events	29
3.2.7	Other Local Events	29
3.3	Receiver Details.....	29
3.3.1	Abstract Data Model	29
3.3.1.1	Receiver Identifier	29
3.3.1.2	List of Incomplete Tasks	30
3.3.1.3	State.....	30
3.3.2	Timers	30
3.3.3	Initialization	30
3.3.4	Higher-Layer Triggered Events.....	30
3.3.5	Message Processing Events and Sequencing Rules.....	30
3.3.5.1	Sending a proc_MSS_PropagationQueryComponentPickUpNewPropagationItems Message	30
3.3.5.2	Receiving a Propagation Tasks Result Set	31
3.3.5.3	Sending a proc_MSS_PropagationQueryComponentReportTaskReady Message	31
3.3.6	Timer Events	31
3.3.7	Other Local Events	31
4	Protocol Examples.....	32
4.1	Component Addition Propagation	32
4.1.1	Initial State	32
4.1.1.1	DB-1	32
4.1.1.1.1	List of Ready Query Components.....	32
4.1.1.1.2	List of Running Tasks.....	32
4.1.1.2	SEN-1	32
4.1.1.2.1	Search Application Name.....	32
4.1.1.2.2	Sender Identifier	32

4.1.1.2.3	List of Ready Query Components	33
4.1.1.2.4	List of Completed Tasks	33
4.1.1.3	REC-1	33
4.1.1.3.1	Receiver Identifier	33
4.1.1.3.2	List of Incomplete Tasks	33
4.1.1.4	REC-2	33
4.1.1.4.1	Receiver Identifier	33
4.1.1.4.2	List of Incomplete Tasks	33
4.1.2	Sequence	33
5	Security	37
5.1	Security Considerations for Implementers	37
5.2	Index of Security Parameters	37
6	Appendix A: Product Behavior	38
7	Change Tracking	39
8	Index	40

1 Introduction

This document specifies the Content Index Propagation Protocol. It is a complete protocol, not an extension of an existing one. The protocol is used to replicate data across multiple servers and to maintain consistency among those servers in the event of changes to that data.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

**little-endian
Unicode**

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

**back-end database server
component
component birth date
crawl component
datetime
document identifier
document set
farm
full-text index catalog
full-text index component
index identifier
index partition
query
query component
query topology
result set
return code
search service application
static rank
stored procedure
Transact-Structured Query Language (T-SQL)**

The following terms are specific to this document:

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

1.2 References

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

1.2.1 Normative References

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

[Iseminger] Microsoft Corporation, "SQL Server 2000 Architecture and XML/Internet Support", Volume 1 of Microsoft SQL Server 2000 Reference Library, Microsoft Press, 2001, ISBN 0-7356-1280-3, <http://www.microsoft.com/mspress/books/5001.aspx>

[MS-CIFO] Microsoft Corporation, "[Content Index Format Structure Specification](#)".

[MSDN-TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference", [http://msdn.microsoft.com/en-us/library/ms189826\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms189826(SQL.90).aspx)

[MS-SRCHTP] Microsoft Corporation, "[Search Topology Protocol Specification](#)".

[MS-TDS] Microsoft Corporation, "[Tabular Data Stream Protocol Specification](#)".

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

1.2.2 Informative References

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

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

[MS-SMB] Microsoft Corporation, "[Server Message Block \(SMB\) Protocol Specification](#)".

1.3 Protocol Overview (Synopsis)

This document specifies communication between a crawl component (the sender) and a query component (the receiver). This protocol only applies to the activity of replicating full-text index catalog data from the sender into the **full-text index catalog** data used by the receiver serving a Microsoft Office SharePoint Server Search Application.

This protocol is used to synchronize changes made to a full-text index catalog from either static rank computation or component addition across receivers. Component addition includes newly crawled content, comprising additions, revisions, and removals. Also, propagation of the static rank computation is necessary as exactly same queries may be routed to different receivers each time they are made, therefore, the static rank computation increases the probability of retrieving similar results across multiple searches.

In a nutshell, senders inform the back-end database server of any changes, while receivers regularly poll back-end database server for timely propagation of changes & updates. On the other hand, receivers inform the back-end database server that they are up-to-date & this information is propagated to senders through the back-end database server.

This protocol specification applies independently to each search application. If there are two or more search applications on a farm, they will all have same requirements for the implementation of this protocol and will be independent of each other.

The following figure shows a high level view of the communication between the servers involved in the process.

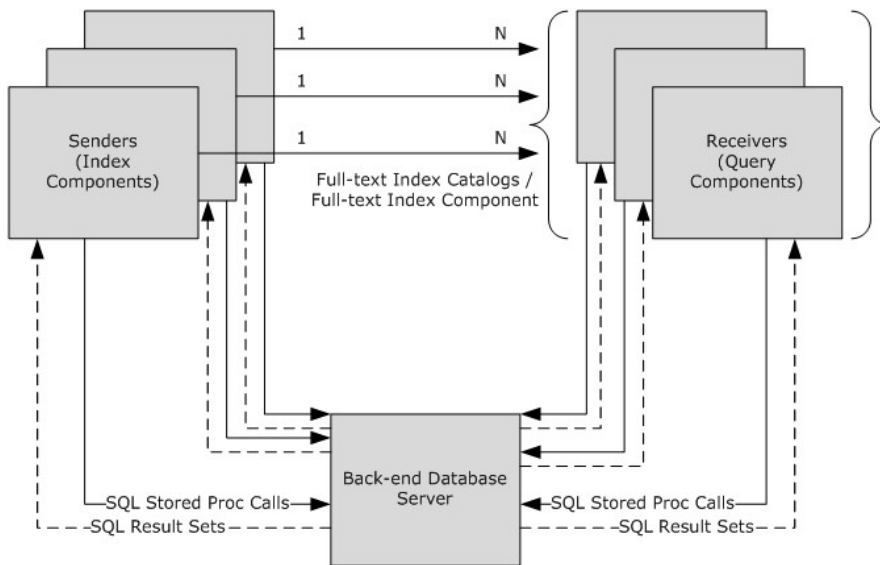


Figure 1: High-level view of communication between servers

The following figure shows the sequence of events during the process of propagation.

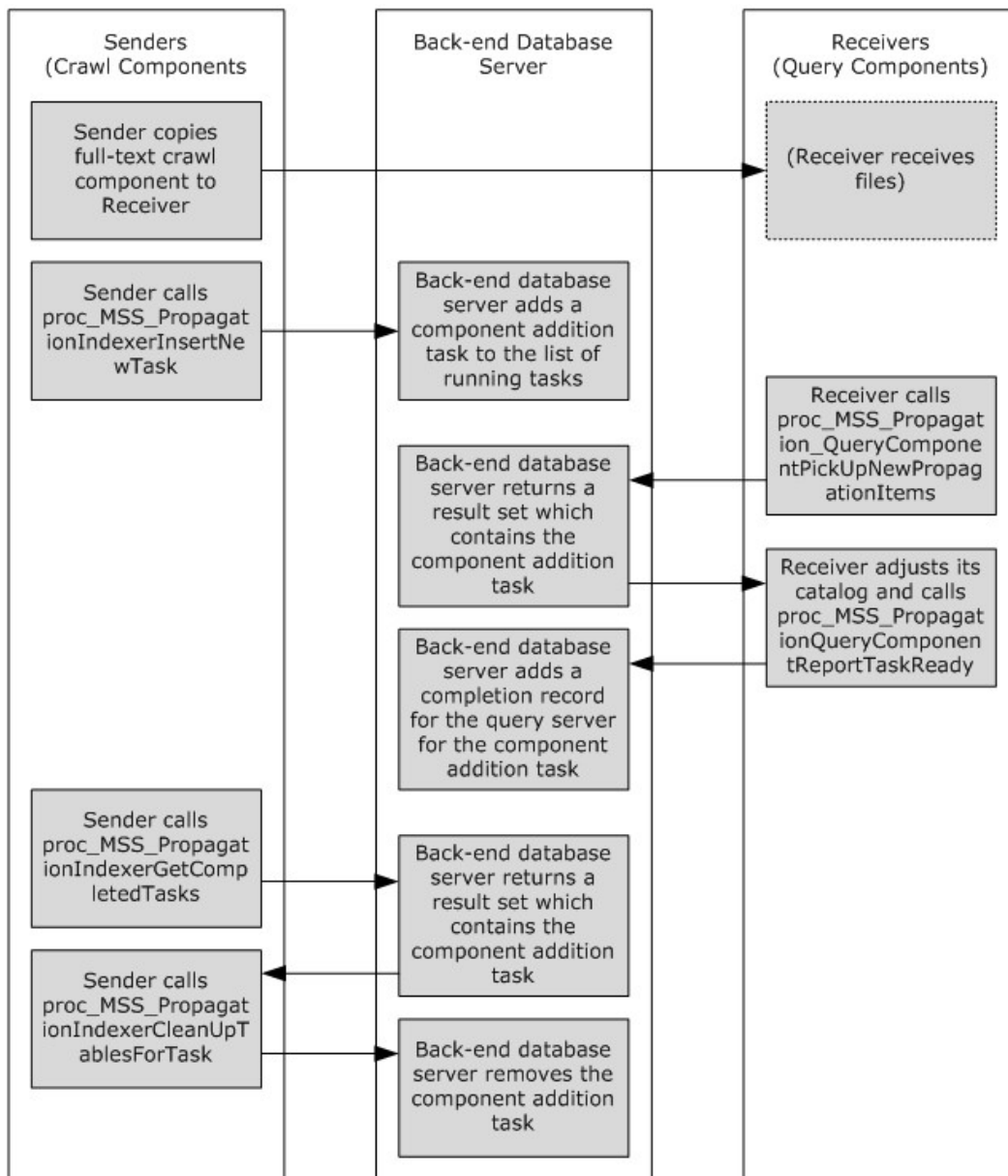


Figure 2: Sequence of operations used to propagate the full-text index component

1.4 Relationship to Other Protocols

The Tabular Data Stream protocol, as described in [\[MS-TDS\]](#), is the transport protocol used to call the **stored procedures**, query SQL views or SQL tables, return **return codes**, and return **result sets**.

This protocol relies on the Server Message Control Block (SMB), as described in [\[MS-SMB\]](#), as its transport protocol to perform server-to-server file copies.

1.5 Prerequisites/Preconditions

This protocol requires that a **farm** be installed and configured. The operations described by the protocol operate between a client that is a part of the farm and a **back-end database server** on which the databases of the farm are stored.

The user that calls the stored procedures specified in this document has permission to read from and write to the databases that contain those stored procedures.

The following prerequisites are also required before the propagation protocol can be successfully invoked. This protocol assumes that the following conditions are true:

- There is a shared directory in the file system on each query server that allows read and write operation by the account named "WSS_WPG" on that query server.
- The stored procedures specified in this document are present on the back-end database server.
- The servers on which the sender and receiver run are members of the farm.

1.6 Applicability Statement

This protocol is applicable only to the activity of replicating full-text index catalog data from a crawl component into the full-text index catalog data used by **query components** serving a Microsoft Office SharePoint Server search application.

1.7 Versioning and Capability Negotiation

Security and Authentication Methods: This protocol supports the SSPI and SQL Authentication with the Protocol Server role described in [\[MS-TDS\]](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Tabular Data Stream protocol, as specified in [\[MS-TDS\]](#), is the transport protocol used to call the stored procedures, query SQL views or SQL tables, return codes, and return result sets.

2.2 Message Syntax

2.2.1 Common Data Types

2.2.1.1 Simple Data Types and Enumerations

2.2.1.2 Enumerations

2.2.1.2.1 Task Type

A task type is a 32-bit signed **integer** used to represent the type of a propagation task. It MUST be one of the values in the following table.

Symbolic name	Value	Description
ComponentAddition	1	A full-text index component will be received by each receiver.
StaticRankComputation	2	All activities performed during a static rank computation event will be performed by each receiver.

2.2.1.2.2 CatalogID

A **CatalogId** is a 32-bit signed **integer** used to represent a full-text index catalog. It MUST be one of the values in the following table.

Value	Description
1	The main catalog, as specified in [MS-CIFO] section 2.18.1.
2	The anchor text catalog, as specified in [MS-CIFO] section 2.18.2.

2.2.2 Full-Text Index Component Message

A full-text index component message is the unit of transfer in full-text index component propagation is a set of files. Each file in this set, except for one, is a duplicate of a file in a full-text index component, as specified in [\[MS-CIFO\]](#) section 2.17, in content, but the extension ".cp" is appended to the original file name to create the name of the duplicate file. Every file of a full-text index component is represented in the set.

The one file in this set that does not correspond to a full-text index component file is a propagation list file, as specified in section [2.2.2.1](#). For an example, see section [4.1](#).

2.2.2.1 Propagation List File

The propagation list file is a list of **Unicode strings** stored in the **string** list format specified in section [2.2.2.1](#). All **integers** and **characters** are stored in **little-endian** form unless specified otherwise.

2.2.2.1.1 String Record

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Number of Characters																															
Characters (variable)																															
...																															

Number of Characters (4 bytes): A 32-bit unsigned **integer** representing the number of characters in the **string**. It MUST be aligned to a 2-byte boundary. It MUST terminate at a 2-byte boundary.

Characters (variable): A variable-length array of 16-bit Unicode values ordered from the beginning to the end of the **string**. It MUST be aligned to a 2-byte boundary. There is no special terminating character. The length of the array is the value of the **Number of Characters** field. It MUST terminate at a 2-byte boundary.

2.2.2.1.2 String List

This is formatted as shown in the following table.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Number of Strings																															
String Records (variable)																															
...																															

Number of Strings (4 bytes): A 32-bit unsigned **integer** representing the number of **strings** in the list. It MUST be located at the beginning of the file.

String Records (variable): A variable-length array of **string** records as specified in section [2.2.2.1.1](#). The number of **string** records in the array is the value of the **Number of Strings** field. It MUST terminate at a 2-byte boundary.

2.2.3 Versioned Index Identifier

This is a 32-bit unsigned **integer** associated with one full-text index component.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved0								Format Version								Reserved1								Index ID							

Reserved0 (1 byte): Must be set to "0x00".

Format Version (1 byte): An 8-bit unsigned **integer** value that MUST be "0x54" if the format version of the full-text index component, as specified in [\[MS-CIFO\]](#) section 2.17, is "0x54". In all other cases, the value MUST be "0x01".

Reserved1 (1 byte): Must be set to "0x00".

Index ID (1 byte): An 8-bit unsigned **integer** equal to the **index identifier** of the full-text index component, as specified in [\[MS-CIFO\]](#) section 2.17.

3 Protocol Details

There are three roles of this protocol:

- Back-end database server
- Sender
- Receiver

Most of the messages sent and received in this protocol are stored procedure calls and the result sets they return. These sproc and result set messages are specified in section [3.1](#). There is one non-sproc-related message, a file transfer from sender to receiver, that is specified in section [2.1](#).

3.1 Back-End Database Server Details

The back-end database server responds to stored procedure calls from the sender and the receiver. It returns result sets and return codes and never initiates communication with the other endpoints of the protocol.

3.1.1 Abstract Data Model

The following section specifies data and state that are sufficient to specify the behavior of the back-end database server. The only state necessary for execution of this protocol from the back-end database server is a list of ready query components, as specified in section [3.1.1.1](#), and a list of running tasks, as specified in section [3.1.1.2](#).

3.1.1.1 List of Ready Query Components

A list of zero or more query components that includes all of the query components in the query component set, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3, that meet the following criteria:

- **State** value, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3, is equal to any of the following, as specified in [\[MS-SRCHTP\]](#) section 2.2.1.3:
 - Ready
 - SplittingIndexes
 - Reverting
 - IndexSplitDone
- Are members of the **query topology** in which the value of **State** is "Active", as specified in [\[MS-SRCHTP\]](#) section 2.2.1.2.

The list does not contain any other query components.

3.1.1.2 List of Running Tasks

A list of zero or more running tasks. Each running task represents one propagation task that is currently being performed by all query components. A running task has the following properties:

taskType: The task type, as specified in section [2.2.1.2.1](#), of the propagation task.

senderID: The **CrawlComponentNumber**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3, of the **crawl component** that created the propagation task.

catalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog to which the propagation task applies.

list of completions: A list of zero or more query components that have completed the propagation task.

For component additions, a running task also has the following properties:

componentID: The versioned index identifier, as specified in section [2.2.3](#), of the full-text index component being added.

maxDocID: The maximum **document identifier** of the full-text index component being added.

birthDate: The **component birth date** of the full-text index component being added.

3.1.2 Timers

None.

3.1.3 Initialization

Listening endpoints are set up on the back-end database server to handle inbound Tabular Data Stream (TDS) requests.

Authentication of the TDS connection to the back-end database server **MUST** occur before this protocol can be used.

The data structures, stored procedures, and actual data are persisted by the back-end database server within databases, so any operations to initialize the state of the database **MUST** occur before the back-end database server can use this protocol. This protocol requires that the search administration data already exists within the back-end database server in a valid state.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The back-end database server does not initiate any communication. It only issues messages to other servers as result sets and return values, in direct response to incoming stored procedure calls.

There are no preconditions of state to receiving any of these calls; the back-end database server **MUST** be able to process them in any order, at any time after initialization.

As an aid to understanding, there is a naming convention for all of the propagation-related stored procedures. Procedures beginning with the prefix "proc_MSS_PropagationIndexer" are called from the sender. Procedures beginning with "proc_MSS_PropagationQueryComponent" are called from the receiver.

3.1.5.1 proc_MSS_PropagationIndexerCleanUpTablesForTask

The **proc_MSS_PropagationIndexerCleanUpTablesForTask** stored procedure is called to remove all records related to a completed propagation task. The **T-SQL** syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationIndexerCleanUpTablesForTask (
    @SenderID          int,
    @CatalogID        int,
    @TaskType          int,
    @ObjectID          int
);
```

@SenderID: The sender identifier, as specified in section [3.2.1.2](#), of the sender that created the propagation task.

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), representing the full-text index catalog to which the propagation task applies.

@TaskType: The task type, as specified in section [2.2.1.2.1](#), of the propagation task.

@ObjectID: If *@TaskType* is "ComponentAddition", as specified in section [2.2.1.2.1](#), *@ObjectID* MUST be either zero or the versioned index identifier, as specified in section [2.2.3](#), of the full-text index component that was propagated. If *@TaskType* is "StaticRankComputation", as specified in section [2.2.1.2.1](#), *@ObjectID* MUST be zero.

When the back-end database server receives this message:

- If **State**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3, of the crawl component that created the propagation task is either "Disabled" or "DisableForRemove", as specified in [\[MS-SRCHTP\]](#) section 2.2.1.7, it MUST do nothing.
- Otherwise, it MUST remove any propagation task from the list of running tasks where **senderID** equals *@SenderID*, **catalogID** equals *@CatalogID*, **taskType** equals *@TaskType*, and **objectID** equals *@ObjectID*. These parameters are specified in section [3.1.1.2](#).

Return Code Values:

Value	Description
0	The task was added to the list of running tasks.
1	The task was not added to the list of running tasks because the crawl component was disabled.

Result Sets: MUST NOT return any result sets.

3.1.5.2 proc_MSS_PropagationIndexerGetCompletedTasks

The **proc_MSS_PropagationIndexerGetCompletedTasks** stored procedure is called to retrieve every propagation task for a specified full-text index catalog that has been completed by all query components in the list of ready query components, as specified in section [3.1.1.1](#). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationIndexerGetCompletedTasks (
    @SenderID          int,
```



```

    @CatalogID          int
);

```

@SenderID: The sender identifier, as specified in section [3.2.1.2](#), of the sender that created the propagation task.

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog for which the caller receives completed propagation tasks.

Return Code Values: An **integer** that MUST be zero.

Result Sets: MUST return the result set specified in section [3.1.5.2.1](#).

- If the value of **State**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3, of the crawl component that created the propagation task is either "Disabled" or "DisableForRemove", as specified in [\[MS-SRCHTP\]](#) section 2.2.1.7, the returned result set MUST contain zero results.
- Otherwise, the returned result set MUST contain exactly one result for each propagation task that has been completed by all query components in the list of ready query components, as specified in section [3.1.1.1](#), and MUST NOT contain other results.

3.1.5.2.1 Completed Tasks Result Set

The T-SQL syntax for the result set is as follows:

```

SenderID          int,
CatalogID        int,
TaskType         int,
ObjectID         int,
{MaxWorkID}      int,
{BirthDate}      int

```

SenderID: The **senderID**, as specified in section [3.1.1.2](#), of the propagation task.

CatalogID: The **catalogID**, as specified in section [3.1.1.2](#), of the propagation task.

TaskType: The **taskType**, as specified in section [3.1.1.2](#), of the propagation task.

ObjectID: If **taskType**, as specified in section [3.1.1.2](#), is **ComponentAddition**, as specified in section [2.2.1.2.1](#), **ObjectID** MUST be the **componentID**, as specified in section [3.1.1.2](#), of the running propagation task. If **taskType**, as specified in section [3.1.1.2](#), is "StaticRankComputation", as specified in section [2.2.1.2.1](#), **ObjectID** MUST be zero.

{MaxWorkID}: MUST be zero.

{BirthDate}: MUST be zero.

3.1.5.3 proc_MSS_PropagationIndexerGetReadyQueryComponents

The **proc_MSS_PropagationIndexerGetReadyQueryComponents** stored procedure is called to retrieve information about all query components in the list of ready query components, as specified in section [3.1.1.1](#). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_MSS_PropagationIndexerGetReadyQueryComponents();

```

Return Code Values: An **integer** that MUST be zero.

Result Sets: MUST return exactly one ready query components result set, as specified in section [3.1.5.3.1](#). This result set MUST contain exactly one result for each query component in the list of ready query components, as specified in section [3.1.1.2](#), and MUST NOT contain other results.

3.1.5.3.1 Ready Query Components Result Set

The T-SQL syntax for the result set is as follows:

```
ServerName          nvarchar(256),
QueryComponentNumber int,
PartitionID         uniqueidentifier,
ShareName           nvarchar(260)
```

ServerName: The **ServerName**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the query component.

QueryComponentNumber: The **QueryComponentNumber**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the query component.

PartitionID: The **PartitionID**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the query component.

ShareName: The **ShareName**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the query component.

3.1.5.4 proc_MSS_PropagationIndexerGetTasks

The **proc_MSS_PropagationIndexerGetTasks** stored procedure is called to retrieve every propagation task that was created by the calling crawl component for a specified full-text index catalog. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationIndexerGetCompletedTasks (
    @SenderID          int,
    @CatalogID         int
);
```

@SenderID: The sender identifier, as specified in section [3.2.1.2](#), of the calling sender.

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog for which the caller receives propagation tasks.

Return Code Values: An **integer** that MUST be zero.

Result Sets: MUST return a propagation tasks result set, as specified in section [3.1.5.6.1](#). The returned result set MUST contain exactly one result for each propagation task in the list of running tasks, as specified in section [3.1.1.2](#), and MUST NOT contain other results.

3.1.5.4.1 Propagation Tasks Result Set

The T-SQL syntax for the result set is as follows:

```
SenderID          int,
```

```

CatalogID          int,
TaskType           int,
ObjectID           int,
{MaxWorkID}       int,
{BirthDate}       int

```

SenderID: The **senderID**, as specified in section [3.1.1.2](#), of the propagation task.

CatalogID: The **catalogID**, as specified in section [3.1.1.2](#), of the propagation task.

TaskType: The **taskType**, as specified in section [3.1.1.2](#), of the propagation task.

ObjectID: If **taskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), **ObjectID** MUST be the **componentID**, as specified in section [3.1.1.2](#), of the running propagation task. If **taskType** is "StaticRankComputation", as specified in section [2.2.1.2.1](#), **ObjectID** MUST be zero.

{MaxWorkID}: MUST be zero.

{BirthDate}: MUST be zero.

3.1.5.5 **proc_MSS_PropagationIndexerInsertNewTask**

The **proc_MSS_PropagationIndexerInsertNewTask** stored procedure is called to add a new propagation task to the list of running tasks (section [3.1.1.2](#)). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_MSS_PropagationIndexerInsertNewTask(
    @SenderID          int,
    @CatalogID        int,
    @TaskType          int,
    @ObjectID          int,
    @MaxWorkID         int,
    @BirthDate         int
);

```

@SenderID: The sender identification, as specified in section [3.2.1.2](#), of the calling sender.

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog to which the propagation task applies.

@TaskType: The task type, as specified in section [2.2.1.2.1](#), of the propagation task.

@ObjectID: If **@TaskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), **@ObjectID** MUST be either zero or the versioned index identifier, as specified in section [2.2.3](#), of the full-text index component that is being propagated. If **@TaskType** is "StaticRankComputation", as specified in section [2.2.1.2.1](#), **@ObjectID** MUST be zero.

@MaxWorkID: If **@TaskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), **@MaxWorkID** MUST be either zero or the maximum document identifier in the full-text index component being propagated. If **@TaskType** is not "ComponentAddition", as specified in section [2.2.1.2.1](#), **@MaxWorkID** MUST be zero.

@BirthDate: If **@TaskType** is "ComponentAddition", **@BirthDate** MUST be either zero or the component birth date of the full-text index component being propagated. If **@TaskType** is not "ComponentAddition", **@BirthDate** MUST be zero.

When the back-end database server receives this message:

- If the value of **State**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3, of the crawl component that created the propagation task is either "Disabled" or "DisableForRemove", as specified in [\[MS-SRCHTP\]](#) section 2.2.1.7, the back-end database server MUST return 1.
- Otherwise,
 1. If the *@CatalogID*, *@TaskType*, and *@ObjectID* parameters match the **catalogID**, **taskType**, and **objectID** values of a propagation task in the list of running tasks, as specified in section [3.1.1.2](#), the back-end database server MUST return 1.
 2. Otherwise, the back-end database server MUST add a new propagation task to the list of running tasks, where **catalogID**, as specified in section [3.1.1.2](#), equals *@CatalogID*, **taskType**, as specified in section [3.1.1.2](#), equals *@TaskType*, **objectID**, as specified in section [3.1.1.2](#), equals *@ObjectID*, and **startTime**, as specified in section [3.1.1.2](#), equals the current local time in **datetime** format.

Return Code Values:

Value	Description
0	The propagation task was added.
1	No propagation task was added because the crawl component was disabled or a duplicate propagation task already existed.

Result Set: MUST NOT return any result set.

3.1.5.6 **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems**

The **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** stored procedure is called to get information about all the running tasks for a particular full-text index catalog that have not yet been completed by a particular query component. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationQueryComponentPickUpNewPropagationItems (  
    @CatalogID          int,  
    @ReceiverID         int  
);
```

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), representing the full-text index catalog to which the retrieved propagation tasks apply.

@ReceiverID: The receiver identifier, as specified in section [3.3.1.1](#), of the calling receiver.

When the back-end database server receives this message:

- If no query component with *@ReceiverID* is in the list of ready query components, as specified in section [3.1.1.1](#), it MUST return 1.
- Otherwise, it MUST return zero.

Return Code Values:

Value	Description
0	A result set with the incomplete propagation tasks was returned.
1	No result set was returned because the query component was not in the list of ready query components.

Result Sets:

- If no query component with *@ReceiverID* is in the list of ready query components, the back-end database server MUST NOT return any result sets.
- Otherwise, it MUST return exactly one Propagation Tasks result set, as specified in section [3.1.5.6.1](#). The result set MUST include exactly one result for each incomplete propagation task for the full-text index catalog identified by *@CatalogID*. A propagation task is incomplete if it has not been reported complete by at least one of the query components in the list of ready query components, as specified in section [3.1.1.1](#). The result set MUST NOT include any other results. The results MUST be ordered in ascending order, primarily by **senderID** values, as specified in section [3.1.5.6.1](#), and secondarily by **birthDate** values, as specified in section [3.1.5.6.1](#).

3.1.5.6.1 Propagation Tasks Result Set

The T-SQL syntax for the result set is as follows:

```

SenderID          int,
CatalogID        int,
TaskType          int,
ObjectID         int,
MaxWorkID        int,
BirthDate        int

```

SenderID: The **senderID**, as specified in section [3.1.1.2](#), of the propagation task.

CatalogID: The **catalogID**, as specified in section [3.1.1.2](#), of the propagation task. This MUST be the same value as the input parameter *@CatalogID*.

TaskType: The **taskType**, as specified in section [3.1.1.2](#), of the propagation task.

ObjectID: For all results where the value of **taskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), **ObjectID** MUST be the **componentID**, as specified in section [3.1.1.2](#), of the propagation task. For all results where the value of **taskType** is not "ComponentAddition", **ObjectID** MUST be zero.

MaxWorkID: For all results where the value of **TaskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), **MaxWorkID** MUST be the value of **maxDocID**, as specified in section [3.1.1.2](#), of the propagation task. For all results where the value of **TaskType** is not "ComponentAddition", **MaxWorkID** MUST be zero.

BirthDate: For all results where the value of **TaskType** is "ComponentAddition", **BirthDate** MUST be the value of **birthdate**, as specified in section [3.1.1.2](#), of the propagation task. For all results where the value of **TaskType** value is not "ComponentAddition", **BirthDate** MUST be zero.

3.1.5.7 proc_MSS_PropagationIndexerDeleteAllTasksFromSender

The **proc_MSS_PropagationIndexerDeleteAllTasksFromSender** stored procedure is called to delete all propagation tasks from the list of running tasks, as specified in section [3.1.1.2](#), that were created by the calling sender. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationIndexerDeleteAllTasksFromSender (
    @SenderID          int,
    @CatalogID        int
);
```

@SenderID: The sender identifier, as specified in section [3.2.1.2](#), of the calling sender.

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog for which the propagation tasks are removed from the list of running tasks, as specified in section [3.1.1.2](#).

Return Code Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result set.

3.1.5.8 proc_MSS_PropagationIndexerGetActiveIndexPartitionGuids

The **proc_MSS_PropagationIndexerGetActiveIndexPartitionGuids** stored procedure is called to get information about the **index partitions** in the active query topology. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationIndexerGetActiveIndexPartitionGuids;
```

Return Code Values: An **integer** that MUST be zero.

Result Sets: MUST return a partitions result set, as specified in section [3.1.5.8.1](#). It MUST contain a result for each index partition, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, in the query topology in the query topology set, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, in which the value of **State**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, is "Active", as specified in [\[MS-SRCHTP\]](#) section 2.2.1.2.

3.1.5.8.1 Partitions Result Set

The T-SQL syntax for the result set is as follows:

```
Ordinal          tinyint,
PartitionID      uniqueidentifier
```

Ordinal: The ordinal value, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the index partition.

PartitionID: The identifier value, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the index partition.

3.1.5.9 proc_MSS_PropagationIndexerGetActiveIndexPartitionHashes

The **proc_MSS_PropagationIndexerGetActiveIndexPartitionHashes** stored procedure is called to get information about the query topology in the active query topology. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationIndexerGetActiveIndexPartitionHashes;
```

Return Code Values: An **integer** that MUST be zero.

Result Sets: MUST return a partition hashes result set, as specified in section [3.1.5.9.1](#). It MUST contain a result for each item in the index partition hash set, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the query topology in the query topology set, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, in which the value of **State**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, is "Active", as specified in [\[MS-SRCHTP\]](#) section 2.2.1.2. It MUST NOT contain any other results.

3.1.5.9.1 Partition Hashes

The T-SQL syntax for the result set is as follows:

Hash	tinyint,
Ordinal	tinyint

Hash: A value between zero and 255.

Ordinal: The value of **Ordinal**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2) of the index partition to which the value of **Hash** is associated in the index partition hash set, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2.

3.1.5.10 proc_MSS_PropagationQueryComponentReportTaskReady

The **proc_MSS_PropagationQueryComponentReportTaskReady** stored procedure is called to record that a query component has finished processing a propagation task. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PropagationQueryServerReportTaskReady(  
    @SenderID          int,  
    @CatalogID        int,  
    @ReceiverID        int,  
    @TaskType          int,  
    @ObjectID          int  
);
```

@SenderID: The sender identifier, as specified in section [3.2.1.2](#), of the sender that created the propagation task.

@CatalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog to which the propagation task applies.

@ReceiverID: The receiver identifier, as specified in section [3.3.1.1](#), of the calling receiver.

@TaskType: Any value of **taskType**, as specified in section [2.2.1.2.1](#).

@ObjectID: If *@TaskType* is "ComponentAddition", as specified in section [2.2.1.2.1](#), *@ObjectID* MUST be the versioned index identifier, as specified in section [2.2.3](#), of the full-text index component that is being propagated. If *@TaskType* is "StaticRankComputation", as specified in section [2.2.1.2.1](#), *@ObjectID* MUST be zero.

When the back-end database server receives this message, it does the following:

- If there is no query component in the list of ready components, as specified in section [3.1.1.1](#), with **QueryComponentNumber**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, equal to *@ReceiverID*, the back-end database server MUST return 1.
- Otherwise,
 1. If the query component with **QueryComponentNumber** equal to *@ReceiverID* in the list of completions for the propagation task in the list of running tasks where **catalogID** equals *@CatalogID*, **taskType** equals *@TaskType*, and **objectID** equals *@ObjectID*, the back-end database server MUST return 1. The list of completions, list of running tasks, **catalogID**, **taskType**, and **objectID** are specified in section [3.1.1](#).
 2. Otherwise, the back-end database server MUST add the query component with **QueryComponentNumber** equal to *@ReceiverID* to the list of completions for the propagation task in the list of running tasks where **catalogID** equals *@CatalogID*, **taskType** equals *@TaskType*, and **objectID** equals *@ObjectID*. The list of completions, list of running tasks, **catalogID**, **taskType**, and **objectID** are specified in section [3.1.1](#).

Return code values are listed in the following table.

Value	Description
0	Successful execution.
1	No change to the list of running tasks was made, because the receiver was not in the list of ready query components, or a completion for this task was already recorded for the receiver.

Result Set: MUST NOT return any result set.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Sender Details

The sender is implemented by a crawl component. It initiates all propagation sequences.

3.2.1 Abstract Data Model

The following section specifies data and state that are sufficient to specify the behavior of the sender. The data provided explains how the protocol behaves. Implementations do not need to adhere to this model as long as their server-to-server communication is consistent with what is specified in this document.

3.2.1.1 Search Application Name

The name of the **search service application** that the crawl component belongs to.

3.2.1.2 Sender Identifier

An **integer** that uniquely identifies the sender. This MUST be equal to the **CrawlComponentNumber** of the sender, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.3.

3.2.1.3 List of Ready Query Components

A list of zero or more query components to receive full-text index component messages, as specified in section [2.2.2](#). Each ready query component has the following properties:

serverName: The **ServerName** of the query component, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2.

shareName: The **ShareName** of the query component, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2.

3.2.1.4 List of Completed Tasks

A list of zero or more completed tasks. Each completed task has the following properties:

catalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog to which the propagation task applies.

taskType: The task type of the propagation task, as specified in section [2.2.1.2.1](#).

objectID: If **taskType** is "ComponentAddition", the versioned index identifier, as specified in section [2.2.3](#), of the full-text index component being added on the query components. For all other values of **taskType**, the value is not used.

The sender uses this list to record all propagation tasks for which it MUST send **proc_MSS_PropagationIndexerCleanUpTablesForTask** messages.

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

3.2.5.1 Sending a **proc_MSS_PropagationIndexerGetReadyQueryComponents** Message

The sender SHOULD call the **proc_MSS_PropagationIndexerGetReadyQueryComponents** stored procedure, as specified in section [3.1.5.3](#), on a periodic basis. If it does not do this, the

sender MUST use another method of accurately updating its list of ready query components, as specified in section [3.2.1.3](#), to match the back-end database server's list of ready query components, as specified in section [3.1.1.1](#).

3.2.5.2 Receiving a Ready Query Components Result Set

This result set, as specified in section [3.1.5.3.1](#), is received automatically after calling the **proc_MSS_PropagationIndexerGetReadyQueryComponents** stored procedure, as specified in section [3.1.5.2](#). The sender MUST replace its current list of ready query components, as specified in section [3.2.1.3](#), with exactly one ready query component for each received result, where the following are true:

- **serverName**, as specified in section [3.2.1.3](#), is set to the value of **ServerName**, as specified in section [3.1.5.3.1](#), of the result.
- **componentName**, as specified in section [3.2.1.3](#), is set to the value of **QueryComponentName**, as specified in section [3.1.5.3.1](#), of the result.
- **shareName**, as specified in section [3.2.1.3](#), is set to the value of **ShareName**, as specified in section [3.1.5.3.1](#), of the result.

The sender MUST NOT add any other ready query components into its list of ready query components, as specified in section [3.2.1.3](#).

3.2.5.3 Sending a Full-Text Index Component Message

This is the first message of a propagation sequence for a component addition action.

When a full-text index component is generated for a full-text index catalog on the sender, the sender MUST perform the following actions:

1. The sender generates a full-text index component, exactly as specified in [\[MS-CIFO\]](#) section 2.17, except that each file name contains an additional prefix, which must be a 0-prefixed, 4-digit hexadecimal representation of the sender identifier, as specified in section [3.2.1.2](#), plus a period, such as "000A.", and an additional suffix ".cp".
2. The sender generates a propagation list file, as specified in section [2.2.2.1](#), containing the file names (not the full file system paths) of each of the files contained in the propagated full-text index component message other than the propagation list file itself.
3. The sender copies the duplicated full-text index component files and the propagation list file, as specified in section [2.2.2.1](#), to a path relative to **serverName**, as specified in section [3.2.1.3](#), and **shareName**, as specified in section [3.2.1.3](#), of each ready query component in the list of ready query components, as specified in section [3.2.1.3](#)). The destination path SHOULD be `\\<server>\<share>\<application>-query-<receiverID>\Projects\<catalog>\Indexer\CiFiles\<file>`, where the following are true:
 4. `<server>` is the **serverName**, as specified in section [3.2.1.3](#), of the query component.
 5. `<share>` is the **shareName**, as specified in section [3.2.1.3](#), of the query component.
 6. `<application>` is the search application name, as specified in section [4.1.1.2.1](#).
 7. `<receiverID>` is the **QueryComponentNumber**, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2, of the query component.
 8. `<catalog>` is the following:

1. "Portal_Content" if the full-text index catalog is the main catalog, as specified in [\[MS-CIFO\]](#) section 2.18.1.
 2. "AnchorProject" if the full-text index catalog is the anchor text catalog, as specified in [\[MS-CIFO\]](#) section 2.18.2.
9. <file> is the file name specified previously.
10. It MUST then call the **proc_MSS_PropagationIndexerInsertNewTask** stored procedure, as specified in section [3.1.5.5](#).

3.2.5.4 Sending a proc_MSS_PropagationIndexerInsertNewTask Message

This is the first message of the propagation sequence for cleaning (specified below) and static rank computation, and the second message of the propagation sequence for component addition actions.

For any of the following events, the receiver calls the **proc_MSS_PropagationIndexerInsertNewTask** stored procedure, as specified in section [3.1.5.5](#), with the following:

1. Component addition. The sender calls the **proc_MSS_PropagationIndexerInsertNewTask** stored procedure with the following parameters:
 1. @SenderID MUST be the sender identifier, as specified in section [3.2.1.2](#), of this sender.
 2. @CatalogID MUST be the **catalogID**, as specified in section [2.2.1.2.2](#), for the full-text index component.
 3. @TaskType MUST be "ComponentAddition", as specified in section [2.2.1.2.1](#).
 4. @ObjectID MUST be the versioned index identifier, as specified in section [2.2.3](#), of the full-text index component.
 5. @MaxWorkID MUST be the maximum document identifier in the full-text index component.
 6. @BirthDate MUST be the component birth date of the **document set** of the full-text index component.
2. Cleaning. It is often desirable to ensure that all query components have completed all tasks before inserting another one. To accomplish this, the sender SHOULD call the **proc_MSS_PropagationIndexerInsertNewTask** stored procedure with the following parameters:
 1. @SenderID MUST be the sender identifier, as specified in section [3.2.1.2](#), of this sender.
 2. @CatalogID MUST be the **catalogID**, as specified in section [2.2.1.2.2](#), for the full-text index component.
 3. @TaskType MUST be "ComponentAddition", as specified in section [2.2.1.2.1](#).
 4. @ObjectID MUST be zero.
 5. @MaxWorkID MUST be zero.
 6. @BirthDate MUST be zero.

No change in the behavior of the sender is necessary if it does not send this message.

3. Static rank computation. The sender calls the **proc_MSS_PropagationIndexerInsertNewTask** stored procedure with the following parameters:
4. *@SenderID* MUST be the sender identifier, as specified in section [3.2.1.2](#), of this sender.
5. *@CatalogID* MUST be the **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog over which static rank computation is to be performed.
6. *@TaskType* MUST be "StaticRankComputation", as specified in section [2.2.1.2.1](#).
7. *@ObjectID* MUST be zero.
8. *@MaxWorkID* MUST be zero.
9. *@BirthDate* MUST be zero.

3.2.5.5 Sending a **proc_MSS_PropagationIndexerGetCompletedTasks** Message

This stored procedure is called to retrieve information about any propagation tasks in the list of running tasks, as specified in section [3.1.1.2](#), that have been completed by all query components, so that they can be removed the back-end database server's list of running tasks.

The sender MUST call the **proc_MSS_PropagationIndexerGetCompletedTasks** stored procedure, as specified in section [3.1.5.2](#), periodically, for both the main catalog, as specified in [\[MS-CIFO\]](#) section 2.18.1, and the anchor text catalog, as specified in [\[MS-CIFO\]](#) section 2.18.2. The time interval between calls SHOULD be between 3 and 30 seconds, but using another interval does not prevent the successful execution of propagation tasks. The parameters for the stored procedure must be set as follows:

- *@SenderID*, as specified in section [3.1.5.2](#), MUST be the sender identifier, as specified in section [3.2.1.2](#), of this sender.
- *@CatalogID* as specified in section [3.1.5.2](#), MUST be the identifier of the full-text index catalog.

3.2.5.6 Receiving a Completed Tasks Result Set Message

A completed tasks result set, as specified in section [3.1.5.2.1](#), is received automatically following any call to the **proc_MSS_PropagationIndexerGetCompletedTasks** stored procedure, as specified in section [3.1.5.2](#). The full-text index catalog to which the result set applies is evident in the value of **CatalogID**, as specified in section [3.1.5.2.1](#), of each result in the result set. For this full-text index catalog, the sender MUST replace its list of completed tasks, as specified in section [3.2.1.4](#), with a new list containing one completed task for each result in this result set, where the following are true:

- **catalogID**, as specified in section [3.2.1.4](#), is set to the value of **CatalogID**, as specified in section [3.1.5.2.1](#), of the result.
- **taskType**, as specified in section [3.2.1.4](#), is set to the value of **TaskType**, as specified in section [3.1.5.2.1](#), of the result.
- **objectID**, as specified in section [3.2.1.4](#), is set to the value of **ObjectID**, as specified in section [3.1.5.2.1](#), of the result.

3.2.5.7 Sending a **proc_MSS_PropagationIndexerCleanUpTablesForTask** Message

This is the final message sent in the propagation sequence of any propagation task.

Whenever there is at least one propagation task in the list of completed tasks, as specified in section [3.2.1.4](#), the **proc_MSS_PropagationIndexerCleanUpTablesForTask** stored procedure, as specified in section [3.1.5.1](#), MUST be called once for each completed task, using the following parameters:

- *@CatalogID*, as specified in section [3.1.5.1](#), is set to the value of **catalogID**, as specified in section [3.2.1.4](#), of the completed task.
- *@TaskType*, as specified in section [3.1.5.1](#), is set to the value of **taskType**, as specified in section [3.2.1.4](#), of the completed task.
- *@ObjectID*, as specified in section [3.1.5.1](#), is set to the value of **objectID**, as specified in section [3.2.1.4](#), of the completed task.

3.2.5.8 Sending a proc_MSS_PropagationIndexerGetTasks Message

Not part of any sequence.

The sender calls the **proc_MSS_PropagationIndexerGetTasks** stored procedure, as specified in section [3.1.5.4](#), at any time, with no precondition. Processes on the sender use this to get the back-end database server's list of all running tasks, as specified in section [3.1.1.2](#). The *@SenderID* parameter, as specified in section [3.1.5.4](#), MUST be the sender's sender identifier, as specified in section [3.2.1.2](#). The *@CatalogID* parameter, as specified in section [3.1.5.4](#), MUST be the **catalogID**, as specified in section [2.2.1.2.2](#), of a full-text index catalog.

3.2.5.9 Receiving a Propagation Tasks Result Set

This result set is received automatically after calling the **proc_MSS_PropagationIndexerGetTasks** stored procedure, as specified in section [3.2.5.8](#).

Receiving this message MUST NOT affect the state of the sender that is specified in section [3.2.1](#).

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Receiver Details

The receiver is implemented by a query component. A receiver uses the protocol to apply changes to its full-text index catalogs and to perform static rank computation on its full-text index catalogs.

3.3.1 Abstract Data Model

The following section specifies data and state that are sufficient to specify the behavior of the receiver. Implementations do not need to adhere to this model as long as their server-to-server communication is consistent with that which is specified in this document.

3.3.1.1 Receiver Identifier

An **integer** that uniquely identifies the receiver. This MUST be equal to the **QueryComponentNumber** of the query component, as specified in [\[MS-SRCHTP\]](#) section 3.1.1.2.

3.3.1.2 List of Incomplete Tasks

A list of zero or more incomplete tasks. An incomplete task has the following properties:

catalogID: The **catalogID**, as specified in section [2.2.1.2.2](#), of the full-text index catalog to which the propagation task applies.

taskType: The task type of the propagation task, as specified in section [2.2.1.2.1](#).

objectID: If **taskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), the versioned index identifier, as specified in section [2.2.3](#), of the full-text index component being added on the query components. For all other values of **taskType**, the value is not used.

maxWorkID: If **taskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), the maximum document identifier in the full-text index component. For all other values of **taskType**, the value is not used.

birthDate: If **taskType** is "ComponentAddition", as specified in section [2.2.1.2.1](#), the component birth date of the document set of the full-text index component. For all other values of **taskType**, the value is not used.

3.3.1.3 State

None.

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 Sending a **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** Message

All activity on a query component for a propagation sequence begins with this call.

The receiver MUST call the **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** stored procedure, as specified in section [3.1.5.10](#), periodically, for both the main catalog, as specified in [\[MS-CIFO\]](#) section 2.18.1, and the anchor text catalog, as specified in [\[MS-CIFO\]](#) section 2.18.2. The time interval between calls SHOULD be between 3 and 30 seconds, but using another interval does not prevent the successful execution of propagation tasks. The procedure MUST be called with the following parameters:

- *@ReceiverID*, as specified in section [3.1.5.10](#), MUST be the receiver identifier, as specified in section [3.3.1.1](#), of this receiver.
- *@CatalogID*, as specified in section [3.1.5.10](#)) MUST be the identifier of the full-text index catalog.

3.3.5.2 Receiving a Propagation Tasks Result Set

A propagation tasks result set is received automatically after sending a **proc_MSS_PropagationQueryServerPickUpNewPropagationItems** message, as specified in section [3.1.5.10](#). The full-text index catalog to which the result set applies is specified in the value of **CatalogID**, as specified in section [3.1.5.4.1](#), of each result in the result set. For this full-text index catalog, the **query component** MUST replace its list of incomplete tasks, as specified in section [3.3.1.2](#), with a new list that contains one incomplete task for each result in this result set, where the following are true:

- **catalogID**, as specified in section [3.3.1.2](#), is set to the value of **CatalogID**, as specified in section [3.1.5.4.1](#), of the result.
- **taskType**, as specified in section [3.3.1.2](#), is set to the value of **TaskType**, as specified in section [3.1.5.4.1](#), of the result.
- **objectID**, as specified in section [3.3.1.2](#), is set to the value of **ObjectID**, as specified in section [3.1.5.4.1](#), of the result.
- **maxWorkID**, as specified in section [3.3.1.2](#), is set to the value of **MaxWorkID**, as specified in section [3.1.5.4.1](#), of the result.
- **birthdate**, as specified in section [3.3.1.2](#), is the value of **BirthDate**, as specified in section [3.1.5.4.1](#), of the result.

3.3.5.3 Sending a proc_MSS_PropagationQueryComponentReportTaskReady Message

This message is the last message sent by a receiver in the propagation sequence.

The **proc_MSS_PropagationQueryComponentReportTaskReady** stored procedure, as specified in section [3.1.5.10](#), MUST be called once for each incomplete task in the receiver's list of incomplete tasks, as specified in section [3.3.1.2](#), where all of the following are true:

- **@ReceiverID**, as specified in section [3.1.5.10](#), is set to the receiver identifier, as specified in section [3.3.1.1](#).
- **@CatalogID**, as specified in section [3.1.5.10](#), is set to the value of **catalogID**, as specified in section [3.3.1.1](#), of the incomplete task,
- **@TaskType**, as specified in section [3.1.5.10](#), is set to the value of **taskType**, as specified in section [3.3.1.1](#), of the incomplete task.
- **@ObjectID**, as specified in section [3.1.5.10](#), is set to the value of **objectID**, as specified in section [3.3.1.1](#), of the incomplete task.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

4.1 Component Addition Propagation

For the example in the following subsections, a search application is demonstrated that has four actors:

- **DB-1**: a back-end database server.
- **SEN-1**: a sender.
- **REC-1**: a query component.
- **REC-2**: another query component.

4.1.1 Initial State

4.1.1.1 DB-1

4.1.1.1.1 List of Ready Query Components

The list contains two query components:

1. REC-1
 1. **QueryComponentNumber** is zero.
 2. **ServerName** is "REC-1".
 3. **State** is "Ready".
 4. **ShareName** is "4c436ee0-b809-4e8a-b00b-be776306e0ee-query-0".
2. REC-2
 1. **QueryComponentNumber** is "1".
 2. **ServerName** is "REC-2".
 3. **State** is "Ready".
 4. **ShareName** is "4c436ee0-b809-4e8a-b00b-be776306e0ee-query-1".

4.1.1.1.2 List of Running Tasks

The list is empty.

4.1.1.2 SEN-1

4.1.1.2.1 Search Application Name

The name is "4c436ee0-b809-4e8a-b00b-be776306e0ee".

4.1.1.2.2 Sender Identifier

The sender identifier is zero.

4.1.1.2.3 List of Ready Query Components

The list contains two query components:

1. REC-1

1. **receiverID** is zero.
2. **serverName** is "REC-1".
3. **ShareName** is "4c436ee0-b809-4e8a-b00b-be776306e0ee-query-0".

2. REC-2

1. **receiverID** is "1".
2. **serverName** is "REC-2".
3. **shareName** is "4c436ee0-b809-4e8a-b00b-be776306e0ee-query-1".

4.1.1.2.4 List of Completed Tasks

The list is empty.

4.1.1.3 REC-1

4.1.1.3.1 Receiver Identifier

The receiver identifier is zero.

4.1.1.3.2 List of Incomplete Tasks

The list is empty.

4.1.1.4 REC-2

4.1.1.4.1 Receiver Identifier

The receiver identifier is "1".

4.1.1.4.2 List of Incomplete Tasks

The list is empty.

4.1.2 Sequence

Events 1 through 6 are not necessary for the propagation sequence to occur, but are presented to demonstrate the steady state of the system that would be recurring in cycles before the propagation sequence begins in event 7.

1. SEN-1 polls DB-1 every 30 seconds by calling the **proc_MSS_PropagationIndexerGetCompletedTasks** stored procedure with *@CatalogID* set to "1". DB-1 returns zero.
2. DB-1 replies with an empty completed tasks result set, indicating that there are no completed tasks for the main catalog.

3. REC-1 polls DB-1 every 30 seconds by calling the **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** stored procedure with *@CatalogID* set to "1". DB-1 returns zero.
4. DB-1 replies with an empty incomplete tasks result set, indicating that there are currently no propagation tasks for REC-1 to perform.
5. REC-2 polls DB-1 every 30 seconds by calling the **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** stored procedure with *@CatalogID* set to "1". DB-1 returns zero.
6. DB-1 replies with an empty incomplete tasks result set, indicating that there are currently no propagation tasks for REC-2 to perform. At this point the sender has generated a new full-text index component and will propagate the component. The full-text index component has index "ID 0x0001001A", versioned index "ID 0x0054001A", maximum document identifier "471952", and component birth date "414".
7. SEN-1 writes the full-text index component files listed in the following table to both of the following file shares:
 1. \\REC-1\4c436ee0-b809-4e8a-b00b-be776306e0ee-query-0\4c436ee0-b809-4e8a-b00b-be776306e0ee-query-0\Projects\Portal_Content\Indexer\Cifiles
 2. \\REC-2\4c436ee0-b809-4e8a-b00b-be776306e0ee-query-1\4c436ee0-b809-4e8a-b00b-be776306e0ee-query-1\Projects\Portal_Content\Indexer\Cifiles

All file names begin with the **senderID** zero (0), contain one of the file names of a full-text index component, and end with the "cp" extension.

File name
0000.0001001A.ci.cp
0000.0001001A.dir.cp
0000.0001001A.bsi.cp
0000.0001001A.bsd.cp
0000.0001001A.csi.cp
0000.0001001A.csd.cp
0000.0001001A.wid.cp
0000.0001001A.list.cp

1. SEN-1 calls the **proc_MSS_PropagationIndexerInsertNewTask** stored procedure with the following parameters:
 1. *@SenderID* is set to zero ("0").
 2. *@CatalogID* is set to "1".
 3. *@TaskType* is set to "ComponentAddition".
 4. *@ObjectID* is set to "5505050" (hexadecimal equivalent: 0x0054001A).

5. *@MaxWorkID* is set to "471952".
6. *@BirthDate* is set to "414".
2. DB-1 returns zero.
3. REC-1 calls the **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** stored procedure with *@ReceiverID* set to zero and *@CatalogID* set to "1". DB-1 returns zero.
4. DB-1 sends a propagation tasks result set with one result where:
 1. **SenderID** is set to zero.
 2. **CatalogID** is set to "1".
 3. **TaskType** is set to "ComponentAddition".
 4. **ObjectID** is set to "5505050" (hexadecimal equivalent: 0x0054001A).
 5. **MaxWorkID** is set to "471952".
 6. **Birthdate** is set to "414".
5. REC-1 applies the full-text index component and calls the **proc_MSS_PropagationQueryComponentReportTaskReady** stored procedure with the following parameters:
 1. *@ReceiverID* is set to zero.
 2. *@SenderID* is set to zero.
 3. *@CatalogID* is set to "1".
 4. *@TaskType* is set to "ComponentAddition".
 5. *@ObjectID* is set to "5505050".DB-1 returns zero.
6. SEN-1 polls DB-1 again by calling the **proc_MSS_PropagationIndexerGetCompletedTasks** stored procedure with *@CatalogID* set to "1". DB-1 returns zero.
7. DB-1 sends an empty completed tasks result set, indicating that there are no completed tasks for the main catalog.
8. REC-2 calls the **proc_MSS_PropagationQueryComponentPickUpNewPropagationItems** stored procedure with *@ReceiverID* set to "1" and *@CatalogID* set to "1".
9. DB-1 returns the following propagation tasks result set:
 1. **SenderID** is set to zero.
 2. **CatalogID** is set to "1".
 3. **TaskType** is set to "ComponentAddition".
 4. **ObjectID** is set to "5505050"
 5. **MaxWorkID** is set to "471952".

6. **Birthdate** is set to "414".
10. REC-2 applies the full-text index component and calls the **proc_MSS_PropagationQueryComponentReportTaskReady** stored procedure with the following parameters:
 1. *@ReceiverID* is set to "1".
 2. *@SenderID* is set to "1".
 3. *@CatalogID* is set to "1".
 4. *@TaskType* is set to "ComponentAddition".
 5. *@ObjectID* is set to "5505050".DB-1 returns zero.
11. The index server SEN-1 polls DB-1 again by calling the **proc_MSS_PropagationIndexerGetCompletedTasks** stored procedure with *@SenderID* set to zero and *@CatalogID* set to "1". DB-1 returns zero.
12. DB-1 returns the following completed tasks result set:
 1. **SenderID** is set to zero.
 2. **CatalogID** is set to "1".
 3. **TaskType** is set to "ComponentAddition".
 4. **ObjectID** is set to "5505050".
 5. {**MaxWorkID**} is set to zero.
 6. {**Birthdate**} is set to zero.
13. The index server SEN-1 then calls the **proc_MSS_PropagationIndexerCleanUpTablesForTask** stored procedure with the following parameters:
 1. *@SenderID* is set to zero.
 2. *@CatalogID* is set to "1".
 3. *@TaskType* is set to "ComponentAddition".
 4. *@ObjectID* is set to "5505050".DB-1 returns zero.
14. DB-1 deletes the propagation task from its list of running propagation tasks.

5 Security

5.1 Security Considerations for Implementers

Security for this protocol is controlled by the access rights to the databases on the back-end database server, which is negotiated as part of the TDS protocol, as described in [\[MS-TDS\]](#).

To call stored procedures, the sender and receiver runs as an account that has read and write permissions on the back-end database server.

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® Office SharePoint® Server 2007
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2
- Windows® SharePoint® Services 3.0

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
server ([section 3.1.1](#) 14, [section 3.2.1](#) 24,
[section 3.3.1](#) 29)
[Applicability](#) 10

B

[Back-end database interface](#) 14

C

[Capability negotiation](#) 10
[CatalogID simple type](#) 11
[Change tracking](#) 39
Client
[overview](#) 14

D

Data model - abstract
server ([section 3.1.1](#) 14, [section 3.2.1](#) 24,
[section 3.3.1](#) 29)
Data types
[CatalogID simple type](#) 11
[task type simple type](#) 11
Data types - simple
[CatalogID](#) 11
[task type](#) 11

E

Events
local - server ([section 3.1.7](#) 24, [section 3.2.7](#) 29,
[section 3.3.7](#) 31)
timer - server ([section 3.1.6](#) 24, [section 3.2.6](#) 29,
[section 3.3.6](#) 31)
Examples
overview ([section 4](#) 32, [section 4.1](#) 32)
[sequence](#) 33

F

[Fields - vendor-extensible](#) 10
[Full-text index component message](#) 11

G

[Glossary](#) 6

H

Higher-layer triggered events
server ([section 3.1.4](#) 15, [section 3.2.4](#) 25,
[section 3.3.4](#) 30)

I

[Implementer - security considerations](#) 37
[Index of security parameters](#) 37
[Informative references](#) 7
Initialization
server ([section 3.1.3](#) 15, [section 3.2.3](#) 25,
[section 3.3.3](#) 30)

Interfaces - server
[back-end database](#) 14
[receiver](#) 29
[sender](#) 24
[Introduction](#) 6

L

List of incomplete tasks
[server](#) 30
List of ready query components
server ([section 3.1.1.1](#) 14, [section 3.2.1.3](#) 25)
List of running tasks
[server](#) 14
Local events
server ([section 3.1.7](#) 24, [section 3.2.7](#) 29,
[section 3.3.7](#) 31)

M

Message processing
[server](#) 15
Message processing events
[receiving a propagation tasks result set](#) 29
Message processing events - receiver
[receiving a propagation tasks result set](#) 31
[sending a](#)
[proc_MSS_PropagationQueryComponentPickUp](#)
[NewPropagationItems message](#) 30
[sending a](#)
[proc_MSS_PropagationQueryComponentReport](#)
[TaskReady message](#) 31
Message processing events - sender
[receiving a completed tasks result set message](#)
28
[receiving a ready query components result set](#) 26
[sending a full-text-index component message](#) 26
[sending a](#)
[proc_MSS_PropagationIndexerGetCompletedTa](#)
[sks message](#) 28
[sending a](#)
[proc_MSS_PropagationIndexerGetReadyQuery](#)
[Components message](#) 25
[sending a](#)
[proc_MSS_PropagationIndexerGetTasks](#)
[message](#) 29
[sending a](#)
[proc_MSS_PropagationIndexerInsertNewTask](#)
[message](#) 27
Message processing events- sender
[sending a](#)
[proc_MSS_PropagationIndexerCleanUpTablesF](#)
[orTask message](#) 28

Messages

- [full-text index component](#) 11
- [propagation list file](#) 12
- [transport](#) 11

Methods

- [proc MSS PropagationIndexerCleanUpTablesForTask](#) 16
- [proc MSS PropagationIndexerDeleteAllTasksFromSender](#) 22
- [proc MSS PropagationIndexerGetActiveIndexPartitionGuids](#) 22
- [proc MSS PropagationIndexerGetActiveIndexPartitionHashes](#) 23
- [proc MSS PropagationIndexerGetCompletedTasks](#) 16
- [proc MSS PropagationIndexerGetReadyQueryComponents](#) 17
- [proc MSS PropagationIndexerGetTasks](#) 18
- [proc MSS PropagationIndexerInsertNewTask](#) 19
- [proc MSS PropagationQueryComponentPickUpNewPropagationItems](#) 20
- [proc MSS PropagationQueryComponentReportTaskReady](#) 23

N

[Normative references](#) 7

O

[Overview \(synopsis\)](#) 7

P

- [Parameters - security index](#) 37
- [Preconditions](#) 10
- [Prerequisites](#) 10
- [proc MSS PropagationIndexerCleanUpTablesForTask method](#) 16
- [proc MSS PropagationIndexerDeleteAllTasksFromSender method](#) 22
- [proc MSS PropagationIndexerGetActiveIndexPartitionGuids method](#) 22
- [proc MSS PropagationIndexerGetActiveIndexPartitionHashes method](#) 23
- [proc MSS PropagationIndexerGetCompletedTasks method](#) 16
- [proc MSS PropagationIndexerGetReadyQueryComponents method](#) 17
- [proc MSS PropagationIndexerGetTasks method](#) 18
- [proc MSS PropagationIndexerInsertNewTask method](#) 19
- [proc MSS PropagationQueryComponentPickUpNewPropagationItems method](#) 20
- [proc MSS PropagationQueryComponentReportTaskReady method](#) 23
- [Product behavior](#) 38
- [Propagation list file](#) 12

R

Receiver identifier

- [server](#) 29
- [Receiver interface](#) 29
- [References](#) 6
 - [informative](#) 7
 - [normative](#) 7
- [Relationship to other protocols](#) 9

S

- Search application name
 - [server](#) 25
- Security
 - [implementer considerations](#) 37
 - [parameter index](#) 37
- Sender identifier
 - [server](#) 25
- [Sender interface](#) 24
- [Sequence example](#) 33
- Sequencing rules
 - [server](#) 15
- Server
 - abstract data model ([section 3.1.1](#) 14, [section 3.2.1](#) 24, [section 3.3.1](#) 29)
 - [back-end database interface](#) 14
 - [details](#) 24
 - higher-layer triggered events ([section 3.1.4](#) 15, [section 3.2.4](#) 25, [section 3.3.4](#) 30)
 - initialization ([section 3.1.3](#) 15, [section 3.2.3](#) 25, [section 3.3.3](#) 30)
 - [list of completed tasks](#) 25
 - [list of incomplete tasks](#) 30
 - list of ready query components ([section 3.1.1.1](#) 14, [section 3.2.1.3](#) 25)
 - [list of running tasks](#) 14
 - local events ([section 3.1.7](#) 24, [section 3.2.7](#) 29, [section 3.3.7](#) 31)
 - [message processing](#) 15
 - overview ([section 3](#) 14, [section 3.1](#) 14)
 - [proc MSS PropagationIndexerCleanUpTablesForTask method](#) 16
 - [proc MSS PropagationIndexerDeleteAllTasksFromSender method](#) 22
 - [proc MSS PropagationIndexerGetActiveIndexPartitionGuids method](#) 22
 - [proc MSS PropagationIndexerGetActiveIndexPartitionHashes method](#) 23
 - [proc MSS PropagationIndexerGetCompletedTasks method](#) 16
 - [proc MSS PropagationIndexerGetReadyQueryComponents method](#) 17
 - [proc MSS PropagationIndexerGetTasks method](#) 18
 - [proc MSS PropagationIndexerInsertNewTask method](#) 19
 - [proc MSS PropagationQueryComponentPickUpNewPropagationItems method](#) 20
 - [proc MSS PropagationQueryComponentReportTaskReady method](#) 23
 - [receiver identifier](#) 29
 - [receiver interface](#) 29
 - [receiving a completed tasks result set message](#) 28

- receiving a propagation tasks result set ([section 3.2.5.9](#) 29, [section 3.3.5.2](#) 31)
- [receiving a ready query components result set](#) 26
- [search application name](#) 25
- [sender identifier](#) 25
- [sender interface](#) 24
- [sending a full-text-index component message](#) 26
- [sending a
 - \[proc MSS PropagationIndexerCleanUpTablesForTask message\]\(#\) 28](#)
- [sending a
 - \[proc MSS PropagationIndexerGetCompletedTasks message\]\(#\) 28](#)
- [sending a
 - \[proc MSS PropagationIndexerGetReadyQueryComponents message\]\(#\) 25](#)
- [sending a
 - \[proc MSS PropagationIndexerGetTasks message\]\(#\) 29](#)
- [sending a
 - \[proc MSS PropagationIndexerInsertNewTask message\]\(#\) 27](#)
- [sending a
 - \[proc MSS PropagationQueryComponentPickUpNewPropagationItems message\]\(#\) 30](#)
- [sending a
 - \[proc MSS PropagationQueryComponentReportTaskReady message\]\(#\) 31](#)
- [sequencing rules](#) 15
- [state](#) 30
- timer events ([section 3.1.6](#) 24, [section 3.2.6](#) 29, [section 3.3.6](#) 31)
- timers ([section 3.1.2](#) 15, [section 3.2.2](#) 25, [section 3.3.2](#) 30)
- Simple data types
 - [CatalogID](#) 11
 - [task type](#) 11
- [Standards assignments](#) 10
- State
 - [server](#) 30

T

- [Task type simple type](#) 11
- Timer events
 - server ([section 3.1.6](#) 24, [section 3.2.6](#) 29, [section 3.3.6](#) 31)
- Timers
 - server ([section 3.1.2](#) 15, [section 3.2.2](#) 25, [section 3.3.2](#) 30)
- [Tracking changes](#) 39
- [Transport](#) 11
- Triggered events - higher layer
 - server ([section 3.2.4](#) 25, [section 3.3.4](#) 30)
- Triggered events - higher-layer
 - [server](#) 15

V

- [Vendor-extensible fields](#) 10
- [Versioning](#) 10