

[MS-SQLPQ2]: Search Service Database Query Version 2 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
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Minor	Updated the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Minor	Clarified the meaning of the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.5	Minor	Clarified the meaning of the technical content.
04/11/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	7
1.1 Glossary	7
1.2 References	8
1.2.1 Normative References	8
1.2.2 Informative References	9
1.3 Protocol Overview (Synopsis)	9
1.3.1 Query Execution	9
1.3.2 Query Logging Operations	10
1.4 Relationship to Other Protocols	10
1.5 Prerequisites/Preconditions	11
1.6 Applicability Statement	11
1.7 Versioning and Capability Negotiation	11
1.8 Vendor-Extensible Fields	11
1.9 Standards Assignments	11
2 Messages	12
2.1 Transport	12
2.2 Common Data Types	12
2.2.1 Simple Data Types and Enumerations	12
2.2.1.1 Duplicate Identifier Block	12
2.2.1.2 Id Blob	12
2.2.1.3 LastConsumerChangeID	12
2.2.1.4 LastUpdate	12
2.2.1.5 Managed Property Time	13
2.2.1.6 Managed Type	13
2.2.1.7 Scope	13
2.2.1.8 Session Identifier	13
2.2.1.9 Site Collection Identifier	13
2.2.1.10 Site Identifier	13
2.2.2 Bit Fields and Flag Structures	13
2.2.2.1 AdvancedSearch Flag	13
2.2.2.2 Continued Flag	14
2.2.2.3 ResultView Flag	14
2.2.3 Binary Structures	14
2.2.3.1 Alternative Offset Length Pair	14
2.2.3.2 Id With Value structure	15
2.2.3.3 Id Value Pairs Blob	15
2.2.3.4 Offset Length Pair	15
2.2.3.5 Property Values Blob	16
2.2.3.6 Serialized String Values	17
2.2.3.7 Summary Blob	18
2.2.4 Result Sets	20
2.2.4.1 Search Security Descriptor Result Set	20
2.2.5 Tables and Views	20
2.2.5.1 MSSDefinitions	20
2.2.5.2 MSSDocProps	21
2.2.5.3 MSSDocResults	21
2.2.5.4 MSSDocSdids	23
2.2.5.5 MSSQLLogLocation	23
2.2.5.6 MSSQLLogLocationSummary	24

2.2.5.7	MSSQLLogNonClickedUnprocessed	24
2.2.5.8	MSSQLLogUnprocessed	25
2.2.6	XML Structures	26
2.2.6.1	Namespaces	26
2.2.6.2	Simple Types	27
2.2.6.3	Complex Types.....	27
2.2.6.4	Elements	27
2.2.6.5	Attributes	27
2.2.6.6	Groups	27
2.2.6.7	Attribute Groups.....	27
3	Protocol Details.....	28
3.1	Microsoft SharePoint Server 2010 Server Details	28
3.1.1	Abstract Data Model	28
3.1.1.1	Best Bet Set.....	29
3.1.1.2	Detailed Query Log Set.....	29
3.1.1.3	Historical Query Log Set	29
3.1.1.4	Raw Query Log Set	29
3.1.1.5	Search Security Descriptor Set	30
3.1.1.6	Scopes	30
3.1.1.7	Summary Query Log Set	30
3.1.2	Timers	31
3.1.3	Initialization	31
3.1.4	Higher-Layer Triggered Events.....	31
3.1.5	Message Processing Events and Sequencing Rules.....	31
3.1.5.1	proc_MSS_FetchSDs	31
3.1.5.2	proc_MSS_GetDocCountEstimate.....	32
3.1.5.3	proc_MSS_GetDuplicateHashes	32
3.1.5.3.1	Duplicate Identifiers Result Set	33
3.1.5.4	proc_MSS_GetKeywordInformation.....	33
3.1.5.4.1	Keyword Details Result Set.....	34
3.1.5.5	proc_MSS_GetManagedProperties.....	34
3.1.5.5.1	Managed Properties Result Set.....	35
3.1.5.6	proc_MSS_GetManagedPropertyAliases	41
3.1.5.6.1	Managed Property Aliases Result Set	41
3.1.5.7	proc_MSS_GetMultipleResults	41
3.1.5.7.1	Definitions Result Set	42
3.1.5.7.2	High Confidence Result Set.....	43
3.1.5.7.3	Search Common Property Result Set	43
3.1.5.7.4	Search Property Result Set.....	45
3.1.5.8	proc_MSS_GetPronunciations.....	46
3.1.5.8.1	Pronunciations Result Set.....	46
3.1.5.9	proc_MSS_GetQuerySuggestions	46
3.1.5.9.1	Query Suggestions Result Set.....	47
3.1.5.10	proc_MSS_GetRankingModels.....	48
3.1.5.11	proc_MSS_GetSchemaHighLevelInfo	48
3.1.5.11.1	Schema High Level Info Result Set	48
3.1.5.12	proc_MSS_GetSchemaParameters	49
3.1.5.12.1	Schema Parameters Result Set	50
3.1.5.13	proc_MSS_GetStaticRankingFeatures	50
3.1.5.14	proc_MSS_GetVectorI4ConfigurationProperty	50
3.1.5.14.1	VectorI4ConfigurationProperty Result Set.....	50
3.1.5.15	proc_MSS_PrefetchSDs.....	51

3.1.5.16	proc_MSS_QLog_AllocateClickIds.....	51
3.1.5.16.1	Click Id Result Set	52
3.1.5.17	proc_MSS_QLog_InsertQueryInfo	52
3.1.5.18	proc_MSS_ScopeCacheGetChangedConsumers	52
3.1.5.18.1	Changed Consumers Result Set.....	53
3.1.5.18.2	Last Consumer Change Identifier Result Set	53
3.1.5.19	proc_MSS_ScopeCacheGetChanges	53
3.1.5.19.1	Scopes Result Set.....	54
3.1.5.19.2	Display Groups Result Set	54
3.1.5.19.3	Compiled Scopes Result Set.....	54
3.1.5.19.4	Last Update Result Set.....	55
3.1.6	Timer Events	55
3.1.7	Other Local Events	55
3.2	Microsoft SharePoint Server 2010 Client Details.....	55
3.2.1	Abstract Data Model	55
3.2.1.1	Query Result Candidate Document Set	56
3.2.1.2	Result Document Property Set.....	56
3.2.2	Timers	56
3.2.3	Initialization	56
3.2.4	Higher-Layer Triggered Events.....	56
3.2.5	Message Processing Events and Sequencing Rules.....	56
3.2.6	Timer Events	57
3.2.7	Other Local Events	57
3.3	Windows SharePoint Services 4.0 Server Details.....	57
3.3.1	Abstract Data Model	57
3.3.2	Timers	58
3.3.3	Initialization	58
3.3.4	Higher-Layer Triggered Events.....	58
3.3.5	Message Processing Events and Sequencing Rules.....	58
3.3.5.1	proc_MSS_FetchSDs	58
3.3.5.2	proc_MSS_GetDuplicateHashes	58
3.3.5.3	proc_MSS_GetManagedProperties.....	58
3.3.5.4	proc_MSS_GetManagedPropertyAliases	58
3.3.5.5	proc_MSS_GetMultipleResults	59
3.3.5.6	proc_MSS_GetSchemaHighLevelInfo	59
3.3.5.7	proc_MSS_GetSchemaParameters	59
3.3.5.8	proc_MSS_GetVectorI4ConfigurationProperty.....	59
3.3.5.9	proc_MSS_PrefetchSDs	59
3.3.6	Timer Events	59
3.3.7	Other Local Events	59
3.4	Windows SharePoint Services 4.0 Client Details.....	59
3.4.1	Abstract Data Model	59
3.4.2	Timers	59
3.4.3	Initialization	59
3.4.4	Higher-Layer Triggered Events.....	59
3.4.5	Message Processing Events and Sequencing Rules.....	59
3.4.6	Timer Events	60
3.4.7	Other Local Events	60
4	Protocol Examples.....	61
4.1	Query Execution	61
4.2	Query Logging.....	62

5 Security	64
5.1 Security Considerations for Implementers.....	64
5.2 Index of Security Parameters	64
6 Appendix A: Product Behavior	65
7 Change Tracking	66
8 Index	67

1 Introduction

This document specifies the Search Service Database Query Version 2 Protocol, which is the communication sequences that are used by the protocol client (Web server and application server) to perform query and logging commands on the protocol server (back-end database server) in relation to search query operations.

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 [\[MS-GLOS\]](#):

- big-endian**
- Coordinated Universal Time (UTC)**
- cyclic redundancy check (CRC)**
- GUID**
- HRESULT**
- Hypertext Transfer Protocol (HTTP)**
- language code identifier (LCID)**
- NULL GUID**

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

- application server**
- back-end database server**
- best bet**
- binary large object (BLOB)**
- content database**
- contextual search scope**
- crawled property**
- crawled property category**
- datetime**
- default search scope**
- document library**
- federated location**
- full-text index catalog**
- high confidence property**
- high confidence results**
- index server**
- item**
- locale**
- managed property**
- mapping order**
- metadata index**
- metadata schema**
- object model**
- pluggable security authentication**
- property identifier**
- property oriented rank**
- query independent rank**
- query result**

query server
query text
rank
result set
search application
search query
search scope
search scope compilation
search scope consumer
search scope index
search security descriptor
site
site collection
sort order
tenant
token
Transact-Structured Query Language (T-SQL)
Uniform Resource Locator (URL)
user name
vector configuration property

The following terms are specific to this document:

alternate access mapping: A mapping of URLs to Web applications. Incoming alternate access mappings are used to provide multiple URL entry points for the same set of content. Outgoing alternate access mappings are used to ensure that content is rendered in the correct URL context.

compiled search scope: A search scope that is the result of the search scope compilation process.

duplicate: A search result that is identified as having identical or near identical content.

relevant result: A search result that is relevant to a query term based on rank. By default, the higher the rank, the higher the item appears in the query results.

search property mapping: A mapping that defines the relationship between a crawled property and a managed property. See also mapping order.

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>

[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-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](#)".

[MS-SQLPADM2] Microsoft Corporation, "[SQL Administration Version 2 Protocol Specification](#)".

[MS-SQLPGAT2] Microsoft Corporation, "[SQL Gatherer Version 2 Protocol Specification](#)".

[MS-SQLPQ2] Microsoft Corporation, "Search Service Database Query Version 2 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>

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

1.2.2 Informative References

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol specifies the communication between the **application server** and the **back-end database server** used to satisfy requests for **search query, managed property** retrieval operations and search query logging operations.

Two distinct roles or clients are served by the back-end database server in the protocol:

- search query and log role: This role serves the search query requests for a **search application**. This role serves requests for search query execution, managed properties retrieval operations and search query logging operations.
- search query role: This role serves the search query request for a search application. This role only serves requests for search query execution and managed properties retrieval operations. There are no search query logging operations for this role.

1.3.1 Query Execution

This protocol enables a search application to provide methods for retrieving **relevant results** that match a search query. This protocol also enables a search application to provide methods for retrieving managed properties for a list of items from the **metadata index**.

The protocol client initiates search query execution to the back-end database server to retrieve attributes that match the specified T-SQL query such as:

- Item identifiers that contain the unique identifier for each search result.
- Duplicate hashes used to resolve any **duplicates** in search results.
- **Search security descriptors** used to determine which users are allowed to view this particular document.
- Managed properties values for a particular item. These can be used to help users identify what they're looking for; for example, author of an item is a managed property.
- **Best bets** assigned by the admin as correct results for a particular user query.
- **High confidence results** produced per the admin configuration as the best result for that particular user query.

1.3.2 Query Logging Operations

This protocol enables a search application to add information to the search query log. The protocol client logs a search query by getting a list of click identifiers from the protocol server. The client then makes a call to add search query information by using the allocated click identifiers. The set of information that is added for a search query depends on users' actions on the search results page.

The information about the search query is automatically added to the search query log. The following additional information is also added to the search query log if the user clicks on at least one **query result**:

- Information about clicked query result.
- Information about all relevant results displayed before the clicked query result (those with a lower result position). If a best bet or a high confidence result was clicked then no unclicked relevant results are logged.

1.4 Relationship to Other Protocols

The server-to-server protocol uses the Tabular Data Stream Protocol as its transport between the front-end web server and the back-end database server. Clients use **Hypertext Transfer Protocol (HTTP)** to communicate with application server.

This protocol relies on [\[MS-TDS\]](#) as its transport protocol to call stored procedures to inspect and manipulate item properties via result sets and return codes.

This relationship is illustrated in the following figure:

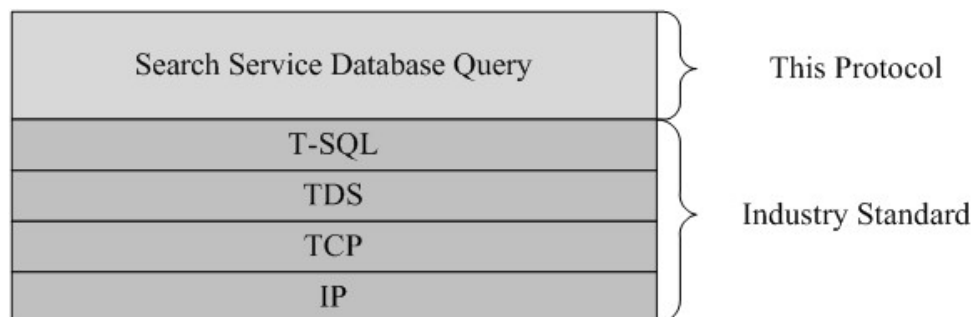


Figure 1: This protocol in relation to other protocols

This protocol relies on [MS-TDS] as its transport protocol to call stored procedures to inspect and manipulate metadata schema via result sets and return codes.

No other protocol depends directly on this protocol.

1.5 Prerequisites/Preconditions

Unless otherwise specified, this protocol requires that the stored procedures and any related data present in the metadata index that is being queried on the back-end database server. The metadata index contains valid data in a consistent state in the order to be queried successfully by the stored procedures.

1.6 Applicability Statement

This protocol is only applicable to application servers when communicating with the back-end database server for search query execution, managed properties retrieval operations and search query logging operations for search query and log role and for search query execution, managed properties retrieval operations for search query role.

1.7 Versioning and Capability Negotiation

Version Negotiation

Versions of the data structures or stored procedures in the database require the same calling parameters and return code values that are expected by the protocol client in order for the stored procedures to be called correctly. If the stored procedures do not provide the same calling parameters or return values as expected, the results of the call are indeterminate. The application server uses stored procedure **proc_MSS_GetDatabaseSchemaVersion** (see [MS-SRCHTP] section 3.1.5.31) to retrieve version of the protocol implemented on the back-end database server and continues using that server only if that version is supported.

This document covers versioning issues in the following areas:

Security and Authentication Methods

This protocol supports the SSPI and SQL Authentication with the back-end database server. These authentication methods are defined in [MS-TDS].

1.8 Vendor-Extensible Fields

This protocol uses **HRESULT** values as defined in [MS-ERREF] section 2.1. Vendors can define their own HRESULT values, provided they set the C bit (0x20000000) for each vendor-defined value, indicating the value is a customer code.

1.9 Standards Assignments

None

2 Messages

2.1 Transport

[\[MS-TDS\]](#) is the transport protocol used to call the stored procedures, query SQL Views or SQL Tables and return codes and result sets.

2.2 Common Data Types

This section contains common definitions used by this protocol.

2.2.1 Simple Data Types and Enumerations

The following subsections define the simple data types for this specification.

2.2.1.1 Duplicate Identifier Block

A 48 byte array of 6 64-bit unsigned integers that identify a portion of an item. Duplication identifiers are used for duplicate result removal if their value is not zero. If two items have the same nonzero duplication identifier then there is a high probability that they are similar.

2.2.1.2 Id Blob

An array of 4 byte integers in **big-endian** format. Each integer represents an identifier.

2.2.1.3 LastConsumerChangeID

Represents a version. When any change is made to the search scope display groups, search scopes or search scope rules of any search scope owned by any search scope consumer, this value is incremented by 1, indicating that a search scope consumer has changed. A search scope consumer is said to have changed in the following scenarios:

- Add / update / delete of a search scope that it owns.
- Add / update / delete of a search scope rule of a search scope that it owns.
- Add / update / delete of a search scope rule that it owns.
- Completion of search scope compilation of a search scope that it owns.

The [LastUpdate](#) data type is incremented every time a specific search scope consumer changes, while the LastConsumerChangeID data type is incremented every time a search scope consumer changes. Thus any search scope consumer whose LastUpdate data type value is equal to the LastConsumerChangeID data type value is the latest search scope consumer to have changed.

2.2.1.4 LastUpdate

An integer value that identifies changes for a **search scope consumer**. Each time a search scope consumer is determined to have changed, the [LastConsumerChangeID](#) is incremented by 1 and the LastUpdate data type of the specific search scope consumer is set to the value of the LastConsumerChangeID. The search scope consumer is said to have changed in the following scenarios:

- Add / update / delete of a search scope that it owns.

- Add / update / delete of a search scope rule of a search scope that it owns.
- Add / update / delete of a search scope rule that it owns.
- Completion of search scope compilation of a search scope that it owns.

2.2.1.5 Managed Property Time

A date/time value in FILETIME format (as defined in [MS-DTYP] section 2.3.1) on which an exclusive logical OR operation is run with 0x8000000000000000 to reverse the high order bit.

2.2.1.6 Managed Type

An integer that identifies the data type of the managed property. The value MUST be in the following table:

Value	Meaning
1	String which is a Unicode character array of arbitrary length
2	64-bit integer
3	64-bit decimal
4	64-bit UTC date/time representing the number of 100-nanosecond intervals since January 1, 1601
5	A 64-bit Boolean value, where -1 represents true and all other values represent false.
6	Binary large object (BLOB)

2.2.1.7 Scope

A list of **search scopes** under which a search query was executed. The search scopes are separated by commas, with whitespace allowed between commas and search scopes.

2.2.1.8 Session Identifier

A **GUID** that is equal to the **NULL GUID** or that is common to a set of search queries.

2.2.1.9 Site Collection Identifier

A GUID that uniquely identifies a **site collection** within a **content database**.

2.2.1.10 Site Identifier

A GUID that uniquely identifies a **site (2)** within a site collection.

2.2.2 Bit Fields and Flag Structures

The following subsections define the bit fields and flag structures for this specification.

2.2.2.1 AdvancedSearch Flag

The AdvancedSearch flag specifies usage of an advanced search page. It MUST be a 1-bit number of type bit.

Value	Meaning
0	The search query was executed from an advanced search page.
1	The search query was not executed from an advanced search page.

2.2.2.2 Continued Flag

The Continued flag identifies the last entry corresponding to a search query in the search query log. It MUST be a 1-bit number of type bit.

Value	Meaning
0	This is the last row of information for this search query in the search query log.
1	This is not the last row of information for this search query in the search query log.

2.2.2.3 ResultView Flag

The ResultView flag identifies the order in which relevant results were ordered. It MUST be an 8-bit number of type tinyint. The ResultView flag bits MUST have one of the values listed in the following table:

Value	Meaning
0	The sort order(1) of relevant results is in descending order of their rank.
1	The sort order(1) of relevant results is in descending order of their managed property modified date.

2.2.3 Binary Structures

The following subsections define the binary structures for this specification.

2.2.3.1 Alternative Offset Length Pair

An Alternative Offset Length Pair augments an Offset Length Pair (as specified in section [2.2.3.4](#)) to specify one or more alternatives to the section of text. The offsets and lengths in this structure refer to an alternative text block, which MAY be different from the text block of the enclosing Offset Length Pair. The alternative text block is not included in the structure.

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		
Offset																Length																	
Next (variable, optional)																																	
...																																	

Offset (2 bytes): An unsigned 16-bit integer. This value modulo 0x8000 is the offset into the alternative text block at which the section described by this Alternative Offset Length Pair begins. This value divided by 0x8000 determines the contents of the **Next** field.

Length (2 bytes): An unsigned 16-bit integer. The value of this field **MUST** be the length of the section described by this Offset Length Pair.

Next (variable, optional): Another Alternative Offset Length Pair structure. This field **MUST** be present if the value of the **Offset** field divided by 0x8000 is greater or equal to 1. Otherwise, it **MUST NOT** be present.

2.2.3.2 Id With Value structure

The following table describes the **Id with Value** structure:

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
Identifier																															
Value																															

Identifier (4 bytes): A 4 byte integer in big-endian format.

Value (4 bytes): A 4 byte integer in big-endian format.

2.2.3.3 Id Value Pairs Blob

The following describes the **Id Value Pairs Blob**:

Id Value Array: An array of Id With Value structures (as specified in section [2.2.3.2](#)) with each element in the array describing an item for which the managed properties are being requested. The format of the element is as follows:

- Identifier: The unique item identifier.
- Value: The corresponding item's **rank**.

2.2.3.4 Offset Length Pair

An Offset Length Pair describes a section of a text block. The text block is not included in the structure.

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
Offset																Length															
Next (variable, optional)																															
...																															

Offset (2 bytes): An unsigned 16-bit integer. This value modulo 0x8000 MUST be the offset into the text block at which the section described by this Offset Length Pair begins. This value divided by 0x8000 determines the contents of the **Next** field.

Length (2 bytes): An unsigned 16-bit integer. This value MUST be the length of the section described by this Offset Length Pair.

Next (variable, optional): An Alternative Offset Length Pair structure (as specified in section [2.2.3.1](#)). This field MUST be present if the value of the **Offset** field divided by 0x8000 is greater or equal to 1. Otherwise, it MUST NOT be present. The sections described by this field MUST be alternatives to the section described by this Offset Length Pair.

2.2.3.5 Property Values Blob

The property values blob contains string and BLOB property values for an item.

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
Version								PropertyCount																Pids (variable)							
...																															
PropertyValueLengths (variable)																															
...																															
PropertyValueCounts (variable)																															
...																															
PropertyValues (variable)																															
...																															

Version (1 byte): 8-bit unsigned integer which MUST be set to the value 2.

PropertyCount (2 bytes): 16-bit unsigned integer which contains the number of properties for which values are stored in the property values blob.

Pids (variable): An array of 32-bit unsigned integers. Each element in this array MUST contain the **property identifier** of the corresponding element in the PropertyValues field. A property identifier MUST NOT appear more than once in the array.

One element in this array MUST exist for each serialized property value in the PropertyValues field at the same ordinal position as the serialized property value. The number of items in this array MUST be equal to the value of the PropertyCount field.

If the value an element in the Pids array is 0x7FFEFFFF then the corresponding blob MUST contain a Summary Blob structure as defined in section [2.2.3.7](#).

PropertyValueLengths (variable): An array of 16-bit unsigned integers. Each element in this array MUST contain the length in bytes of the corresponding element in the PropertyValues field.

One element in this array MUST exist for each serialized property value in the PropertyValues field at the same ordinal position as the serialized property value. The number of items in this array MUST be equal to the value of the PropertyCount field.

PropertyValueCounts (variable): An array of 16-bit unsigned integers. Each element in this array MUST contain

- the value 65535 if the Managed Type (as defined in section [2.2.1.6](#)) of the managed property is 6, or
- the number of values which are serialized for a property in the PropertyValues field otherwise.

One element in this array MUST exist for each serialized property value in the PropertyValues field at the same ordinal position as the serialized property value. The number of items in this array MUST be equal to the value of the PropertyCount field.

PropertyValues (variable): An array of variable-length binary large objects (BLOBs), each of which contains values for a managed property for an item.

For each blob, if the Managed Type (as defined in section [2.2.1.6](#)) of the managed property is 1 then the blob MUST contain a serialized string values structure as defined in section [2.2.3.6](#). Otherwise the Managed Type of the managed property MUST be 6.

The length of each blob in the PropertyValues field MUST be equal to the corresponding array element in the PropertyValueLengths field. The total length of the PropertyValues field MUST be equal to the sum of the lengths in the PropertyValueLengths field.

2.2.3.6 Serialized String Values

The serialized string values structure is an array of bytes which contain a Unicode-encoded string that MUST conform to the following ABNF representation, as specified in [\[RFC5234\]](#).

```

; A sequence is either a single value or multiple values.
; Multiple values always start with a string-delim
; Single values never start with a string-delim
prop-vals-sequence      = string-val / string-vals prop-vals-sequence
; e.g.
; {"a","b"}    -> {"`a`b"}`
; {"a~","~b"} -> {"`a~`~b"}`
; {"a`","`b"} -> {"`a~`~`b"}`
; {"a~b","`"} -> {"`a~b~`~`"}`
string-vals             = string-delim string-val [string-vals]
; e.g.
; {"val"}      -> "val"
; {"a`b"}     -> "a`b"
; {"~`~"}     -> "~`~"
string-val             = char-val [string-val]
char-val              = delimited-backquote / delimited-tilde / unicode-nondelimiter-
char
delimited-tilde       = char-delimiter string-delim
delimited-backquote   = char-delimiter char-delimiter
string-delim          = %x60
```

```

char-delimiter          = %x7E
    ; Unicode code points except for the delimiters
unicode-nondelimiter-char = %x21-5F / %x61-7D / %x7F-10FFFF

```

prop-vals-sequence: the serialized string values structure

string-val: a sequence of characters comprising a string property value.

2.2.3.7 Summary Blob

The Summary Blob contains an excerpt of an **item**. It consists of a part of the text of the item, together with information denoting where tokens in the text begin and end, and where continuous pieces of text, such as paragraphs, begin and end. In addition, it stores information denoting the beginnings and ends of tokens in the title and description of the item. The title and description themselves are not stored.

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		
TitleTokensSize																																	
TitleTokens (variable)																																	
...																																	
DescriptionTokensSize																																	
DescriptionTokens (variable)																																	
...																																	
ContentTokensSize																																	
ContentTokens (variable)																																	
...																																	
ContentChunksSize																																	
ContentChunks (variable)																																	
...																																	
AltTokensSize																																	
AltTokens (variable)																																	
...																																	

ContentsLength
Contents (variable)
...

TitleTokensSize (4 bytes): An unsigned 32-bit integer. The value of the field MUST be the length, in bytes, of the following **TitleTokens** field.

TitleTokens (variable): An array of Offset Length Pair (as specified in section [2.2.3.4](#)), describing the **tokens** in the title of the document. All Offset Length Pairs MUST refer to the same text block. This text block MUST NOT be included in this structure. The alternative text block that the Offset Length Pairs refer to MUST be the contents of the **AltTokens** field.

DescriptionTokensSize (4 bytes): An unsigned 32-bit integer. The value of this field MUST be the length, in bytes, of the following **DescriptionTokens** field.

DescriptionTokens (variable): An array of Offset Length Pair (as specified in section [2.2.3.4](#)), describing the tokens in the description of the document. All Offset Length Pairs MUST refer to the same text block. This text block MUST NOT be included in this structure. The alternative text block that the Offset Length Pairs refer to MUST be the contents of the **AltTokens** field.

ContentTokensSize (4 bytes): An unsigned 32-bit integer. The value of this field MUST be the length, in bytes, of the following **ContentTokens** field.

ContentTokens (variable): An array of Offset Length Pairs (as specified in section [2.2.3.4](#)), describing the tokens in the **Contents** field. The text block that the Offset Length Pairs refer to MUST be the contents of the **Contents** field. The alternative text block that the Offset Length Pairs refer to MUST be the contents of the **AltTokens** field.

ContentChunksSize (4 bytes): An unsigned 32-bit integer. The value of this field MUST be the length, in bytes, of the following **ContentChunks** field.

ContentChunks (variable): An array of Offset Length Pairs (as specified in section [2.2.3.4](#)), describing the chunks in the **Contents** field. The text block that the Offset Length Pairs refer to MUST be the contents of the **Contents** field. There MUST NOT be any Alternative Offset Length Pairs (as specified in section [2.2.3.1](#)) in any of the Offset Length Pairs in this array.

AltTokensSize (4 bytes): An unsigned 32-bit integer. The value of this field MUST be the length, in bytes, of the following **AltTokens** field.

AltTokens (variable): Alternative tokens from the title, description or content of the document. The tokens MUST be stored as a sequence of 16-bit characters, encoded as UTF-16. If there is an odd number of characters to store, another character MUST be added to the sequence. This extra character MUST be ignored.

ContentsLength (4 bytes): An unsigned 32-bit integer. The value of this field MUST be the number of characters in the following **Contents** field.

Contents (variable): The contents of the documents. The words MUST be stored as a sequence of 16-bit characters, encoded as UTF-16.

2.2.4 Result Sets

2.2.4.1 Search Security Descriptor Result Set

The search security descriptor result set contains information about search security descriptors for a list of items. Each row in the result set contains the item identifier and the corresponding search security descriptor. The result set **MUST** contain zero or more rows. The result set **MUST NOT** have more rows than the number of identifiers present in the @joinData parameter.

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

```
sdid          int NOT NULL,  
type         int NOT NULL,  
sd           image NOT NULL;
```

sdid: The identifier of the search security descriptor.

type: The type of the search security descriptor. The mechanism for populating this field is specified in [\[MS-SQLPGAT2\]](#) section 3.1.5.66.

sd: A binary image that contains the search security descriptor of the item. The mechanism for populating this field is specified in [\[MS-SQLPGAT2\]](#) section 3.1.5.66.

2.2.5 Tables and Views

The following subsections define the tables and structures for this specification.

2.2.5.1 MSSDefinitions

The MSSDefinition table stores information about definitions extracted from crawled items.

The T-SQL syntax for the table is as follows:

```
TABLE MSSDefinitions(  
    Term          nvarchar(40) NOT NULL,  
    DocId         int NOT NULL,  
    Sentence      nvarchar(255) NOT NULL,  
    TermOffset    int NOT NULL,  
    TermLength    int NOT NULL,  
    SentenceHash  AS (binary_checksum([Sentence])),  
    IsAcronym     bit NULL  
);
```

Term: The term this definition corresponds to.

DocId: The identifier of an item.

Sentence: The extracted definition of the term.

TermOffset: The character offset at which the specified term appears in **Sentence**.

TermLength: The number of characters in the specified term.

SentenceHash: The checksum of the extracted definition returned by BINARY_CHECKSUM (see [\[MSDN-TSQL-Ref\]](#)).

IsAcronym: MUST be 1 if this definition is an acronym, otherwise MUST be 0.

2.2.5.2 MSSDocProps

The MSSDocProps table stores values for managed properties of items in the metadata index.

The T-SQL syntax for the table is as follows:

```
TABLE MSSDocProps (
    DocId          int NOT NULL,
    Pid            int NOT NULL,
    IVal           bigint NULL,
    StrVal         nvarchar(450) NULL,
    StrVal2       nvarchar(4000) NULL
);
```

DocId: The unique identifier of an item which contains the property whose value is being stored.

Pid: The unique identifier of a managed property whose value is being stored.

IVal: The numeric value of the managed property if the managed type (section [2.2.1.6](#)) of the managed property is equal to 2, 3, 4 or 5. If the managed type (section [2.2.1.6](#)) of the managed property is not equal to 2, 3, 4 or 5, the value is an signature (computed by the client of the protocol) for the value of the managed property. When the type is a date, the value is in Managed Property Time format defined in section [2.2.1.5](#). When the type is a floating point decimal, the value is converted to an integer keeping the number of decimals according the DecimalPlaces attribute of the managed property as specified in [\[MS-SQLPADM2\]](#). The value MUST NOT be NULL.

StrVal: The string value of the managed property when the property type is a string. If the value length is greater than the size allowed by the MaxIndexedStringLength in the managed property definition (as specified in [\[MS-SQLPADM2\]](#)), the string is truncated to MaxIndexedStringLength. StrVal MUST be NULL if the managed type (section [2.2.1.6](#)) of the managed property is not equal to 1 or if the MaxIndexedStringLength property is 0.

StrVal2: The remainder of the string value of the managed property when the property type is a string. If the value length stored in the StrVal column preceding was truncated or NULL, the string overflow is stored in StrVal2 up to the size allowed by the MaxNonIndexedStringLength in the managed property definition as specified in [\[MS-SQLPADM2\]](#). StrVal2 MUST be NULL if the managed type (section [2.2.1.6](#)) of the managed property is not equal to 1 or if the entire length of the string is stored in StrVal or if MaxNonIndexedStringLength is 0.

2.2.5.3 MSSDocResults

The MSSDocResults table stores values for common managed properties of items in the metadata index.

The T-SQL syntax for the table is as follows:

```
TABLE MSSDocResults (
    DocId          int NOT NULL,
    SummaryBlobSize int NULL,
    Size           bigint NULL,
    LastModified   bigint NULL,
    IsDocument     bit NOT NULL,
    IsPictureUrl   bit NOT NULL,
    Author         nvarchar(4000) NULL,
```

```

Title                nvarchar(4000) NULL,
Url                  nvarchar(4000) NULL,
PictureThumbnailUrl nvarchar(4000) NULL,
ContentClass         nvarchar(4000) NULL,
FileExtension        nvarchar(4000) NULL,
Tags                 nvarchar(4000) NULL,
PropertyBlob         nvarbinary(max) NULL,
SummaryBlob          nvarbinary(max) NULL,
PopularSocialTags    nvarchar(4000) NULL,
SiteName             nvarchar(4000) NULL,
Description           nvarchar(4000) NULL,
ParentLink           nvarchar(4000) NULL,
NumberOfMembers      int NULL,
PictureHeightAndWidth bigint NULL,
DisplayDate          bigint NULL
);

```

DocId: The unique identifier of an item which contains the properties whose values are being stored.

SummaryBlobSize: The length in bytes of the summary.

Size: The length in bytes of the document which contains the properties whose values are being stored.

LastModified: **UTC** last modified timestamp as [Managed Property Time](#).

IsDocument: MUST be 1 when the item is a document in a **document library**, MUST be 0 when the item is a list item in a SharePoint list, otherwise MUST be ignored.

IsPictureUrl: MUST be 1 when the value for **PictureThumbnailUrl** is the path to an image otherwise MUST be 0.

Author: The author of the item.

Title: The title of the item.

Url: The path of the item.

PictureThumbnailUrl: The path of an item's thumbnail or the path of an image associated with an item.

ContentClass: A string that describes the type of an item. It MUST be one of the following values:

urn:content-class:SPSSearchQuery, urn:content-class:SPSListing:News, urn:content-class:SPSPeople, urn:content-class:SPSOrganization, urn:content-classes:SPSCategory, urn:content-classes:SPSListing, urn:content-classes:SPSPersonListing, urn:content-classes:SPSTextListing, urn:content-classes:SPSSiteListing, urn:content-classes:SPSSiteRegistry, STS_Web, STS_Site, STS_List_Events, STS_List_Tasks, STS_List_Announcements, STS_List_PictureLibrary, STS_List_DocumentLibrary, STS_ListItem_DocumentLibrary, STS_ListItem_PictureLibrary, STS_List_DiscussionBoard, STS_List_Contacts, STS_List, STS_List_Links, STS_ListItem

FileExtension: The file extension of the item.

Tags: The item's taxonomy terms.

PropertyBlob: A Property values blob structure as defined in section [2.2.3.5](#).

SummaryBlob: This field MUST NOT be used.

PopularSocialTags: The social tags for the item.

SiteName: The name of the item's site (2).

Description: The description of the item.

ParentLink: This field is not used.

NumberOfMembers: The number of members in an organization when the ContentClass is an urn:content-class:SPSOrganization, otherwise MUST be 0.

PictureHeightAndWidth: An 8-byte integer which holds an image's height in the high order DWORD and an image's width in the low order DWORD.

DisplayDate: A UTC date in Managed Property Time which is displayed.

2.2.5.4 MSSDocSdids

The MSSDocSdids table stores identifiers of the search security descriptors of items.

The T-SQL syntax for the table is as follows:

```
TABLE MSSDocSdids (
    DocId                int NOT NULL,
    Type                 smallint NOT NULL,
    Sdid                 int NOT NULL,
    HasPluggableSecurityTrimming bit NOT NULL,
    DuplicateHashes     binary(48) NULL
);
```

DocId: The identifier of the item.

Type: The identifier of the type of the search security descriptor. **Type** MUST be 0 when the search security descriptor of the item is a Windows security descriptor as defined in [\[MS-DTYP\]](#) section 2.4.6. Otherwise, the identifier is not a Windows security descriptor and the **Type** MUST be 1.

Sdid: The unique identifier of the search security descriptor of the item.

HasPluggableSecurityTrimming: Indicates whether item uses **pluggable security authentication**. **HasPluggableSecurityTrimming** MUST be 1 if the item uses pluggable security authentication. Otherwise, **HasPluggableSecurityTrimming** MUST be 0.

DuplicateHashes: A Duplicate Identifier Block as specified in section [2.2.1.1](#).

2.2.5.5 MSSQLLogLocation

The MSSQLLogLocation table stores information about **federated locations**. Each row in this table stores the identifier and name of a federated location.

The T-SQL syntax for the table is as follows:

```
TABLE MSSQLLogLocation (
```

```

        locationId          int NOT NULL,
        location            nvarchar(60) NULL
    );

```

locationId: The identifier of the federated location.

location: The name of the federated location.

2.2.5.6 MSSQLLogLocationSummary

The MSSQLLogLocationSummary table stores daily totals of summary query log information per federated location. The stored procedure `proc_MSS_QLog_InsertQueryInfo` defined in section [3.1.5.17](#) populates this table.

The T-SQL syntax for the table is as follows:

```

TABLE MSSQLLogLocationSummary (
    searchDate            smalldatetime NULL,
    locationId           int NULL,
    numClicks            int NULL,
    numQueries           int NULL
);

```

searchDate: The date on which the search queries were executed.

locationId: The identifier of the federated location where the search query was executed. This value MUST correspond to the **locationId** of one row in the MSSQLLogLocation table defined in section [2.2.5.5](#).

numClicks: The total number of clicked results on the date specified by **searchDate** in the federated location specified by **locationId**.

numQueries: The total number of queries executed on the date specified by **searchDate** in the federated location specified by **locationId**.

2.2.5.7 MSSQLLogNonClickedUnprocessed

The MSSQLLogNonClickedUnprocessed table stores information about non-clicked query results until it is processed and truncated by the stored procedure `proc_MSS_QLog_InsertQueryInfo` defined in section [3.1.5.17](#). Each row in this table corresponds to a non-clicked URI.

The T-SQL syntax for the table is as follows:

```

TABLE MSSQLLogNonClickedUnprocessed (
    clickId              bigint NOT NULL,
    rank                 int NOT NULL,
    nonClickedUrl        nvarchar(1024) NOT NULL
);

```

clickId: The unique identifier of an entry in the search query log.

rank: The rank of the non-clicked query result.

nonClickedUrl: The non-clicked relevant result 's URI.

2.2.5.8 MSSQLLogUnprocessed

The MSSQLLogUnprocessed table stores information about each search query and clicked query results until the information is processed and truncated by the stored procedure `proc_MSS_QLog_InsertQueryInfo` defined in section [3.1.5.17](#). Each row in this table corresponds to either a search query only or clicked query result for a search query.

The T-SQL syntax for the table is as follows:

```
TABLE MSSQLLogUnprocessed (
    clickId                bigint NULL,
    queryString            nvarchar(1024) NULL,
    clickedUrl             nvarchar(1024) NULL,
    resultsUrl            nvarchar(1024) NULL,
    scope                 nvarchar(450) NULL,
    bestBet               nvarchar(100) NULL,
    queryServer           nvarchar(256) NULL,
    siteGuid              uniqueidentifier NULL,
    searchTime            datetime NULL,
    clickTime             datetime NULL,
    searchDate            smalldatetime NULL,
    clickedUrlRank        int NULL,
    numResults            int NULL,
    numHighConf           int NULL,
    numBestBets           int NULL,
    numScopes             int NULL,
    resultView            tinyint NULL,
    advancedSearch        bit NULL,
    didYouMean            nvarchar(256) NULL,
    continued             bit NULL,
    contextualScope       nvarchar(100) NULL,
    contextualScopeUrl   nvarchar(1024) NULL,
    location              nvarchar(60) NULL,
    lcid                  int NULL,
    sessionId             UNIQUEIDENTIFIER NULL,
    userName              nvarchar(256) NULL,
    tenantId              UNIQUEIDENTIFIER NULL,
    definitions           bit NULL
);
```

clickId: This is used to uniquely identify an entry in search query log.

queryString: The **query text** that was executed.

clickedUrl: The query result URI that was clicked. This MUST be NULL if search query information was logged in response to user navigating away from the search results page.

resultsUrl: The URI of the search results page.

scope: The search scopes, under which this search query was executed. The value MUST be a scope data type, as specified in section [2.2.1.7](#).

bestBet: The title of the best bet that was clicked. This MUST be NULL if the clickedUrl is not a best bet URI.

queryServer: The name of the **query server** in which this search query was executed.

siteGuid: The site collection from which the search query was executed. This MUST be a site (2) identifier (or a site collection identifier), as specified in section [2.2.1.9](#).

searchTime: The **datetime** at which this search query was executed.

clickTime: The datetime at which the clickedUrl was clicked in the search results page. This MUST be NULL if clickedURL is NULL.

searchDate: The datetime on which the search query was executed.

clickedUrlRank: The rank of the clicked query result. This MUST be NULL if the clickedUrl is not a search result URI.

numResults: The number of relevant results returned for this search query.

numHighConf: The number of high confidence results returned for this search query.

numBestBets: The number of best bets returned for this search query.

numScopes: The number of search scopes under which this search query was executed. This MUST be equal to the number of search scope names concatenated in the scope column.

resultView: The value MUST be a result view flag, as specified in section [2.2.2.3](#).

advancedSearch: The value MUST be an advanced search flag, as specified in section [2.2.2.1](#).

didYouMean: The value MUST be the spelling suggestion returned for this search query.

continued: The value MUST be a continued flag, as specified in section [2.2.2.2](#).

contextualScope: The **contextual search scope** under which this search query was executed.

contextualScopeUrl: The URI of the **contextualScope**.

location: The name of the federated location where the search query was executed.

lcid: The **language code identifier (LCID)** of the **locale** in which this query was executed.

sessionId: The session in which this query was executed. This value MUST be a session identifier as specified in section [2.2.1.8](#).

userName: The **user name** of the user who initiated the execution of this query.

tenantId: The identifier of the **tenant** that executed the search query.

definitions: The value MUST be 0 if the query returned no definitions, otherwise the value MUST be 1.

2.2.6 XML Structures

No common XML Structures are defined in this protocol.

2.2.6.1 Namespaces

None.

2.2.6.2 Simple Types

None.

2.2.6.3 Complex Types

None.

2.2.6.4 Elements

None.

2.2.6.5 Attributes

None.

2.2.6.6 Groups

None.

2.2.6.7 Attribute Groups

None.

3 Protocol Details

3.1 Microsoft SharePoint Server 2010 Server Details

The search query and log server role is described in this section. This role serves requests for search query execution, managed properties retrieval operations and search query logging operations.

3.1.1 Abstract Data Model

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

The protocol server maintains the following set of metadata for this protocol. Data within the protocol server are maintained until updated or removed.

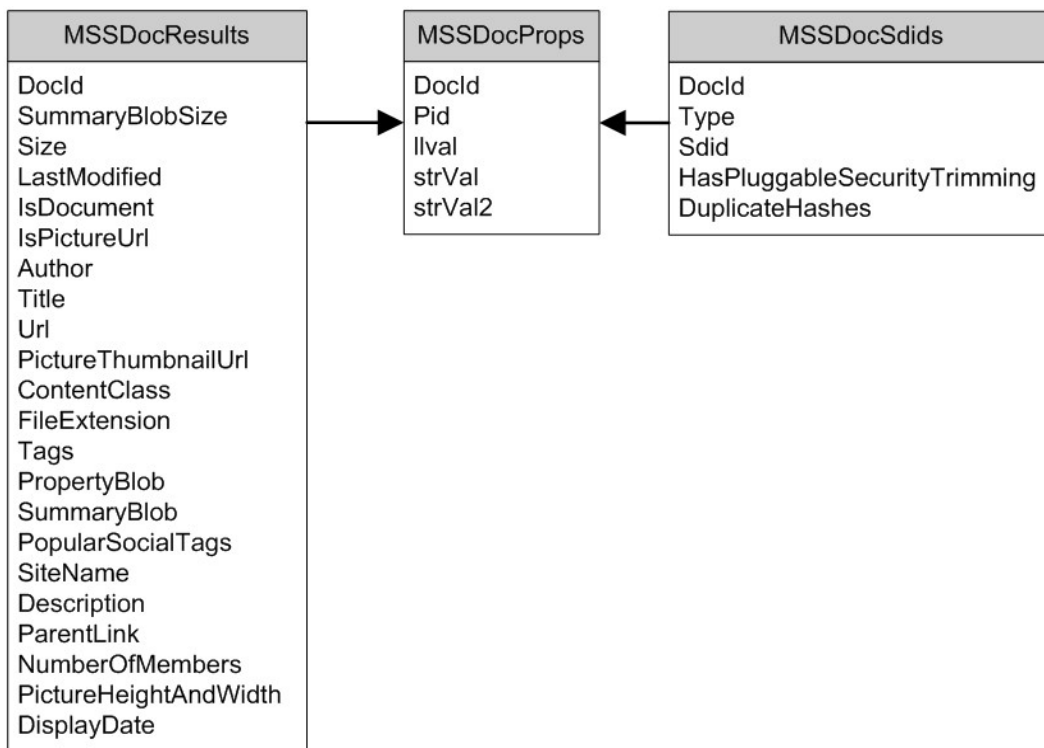


Figure 2: Metadata maintained by a Microsoft SharePoint Server 2010 to support this protocol

The following sections describe these query logging objects:

- Best bet set
- Search security descriptor set
- Scopes

- Raw query log set
- Historical query log set
- Detailed query log set
- Summary query log set

3.1.1.1 Best Bet Set

The Best Bets and Keywords Abstract Data Model is specified in [\[MS-SQLPADM2\]](#) section 3.1.1.2.

3.1.1.2 Detailed Query Log Set

The detailed query log set maintains detailed information about each executed query. The detailed query log set has the following attributes:

- Query text: The query text of the search query that was executed.
- Site GUID: The GUID of the site (2) (or of the site collection) from which the search query was executed.
- Search scope: The search scopes, under which this search query was executed.
- Contextual Search scope: The contextual search scope under which this search query was executed.
- Results Page URL: The URI of the search results page.
- Best bets: The best bets that were clicked for the query text.
- Query Server: The name of the query server in which this search query was executed.
- Federated location: The name of the federated location where the search query was executed.

3.1.1.3 Historical Query Log Set

The historical query log set stores aggregated information about the number of queries executed on each day. The historical query log set has the following attributes:

- Search date: The date on which at least one search query was executed.
- Number of queries executed on search date.
- Site GUID: The GUID of the site (2) (or of the site collection) from which the search query was executed.

3.1.1.4 Raw Query Log Set

The raw query log set contains all the information about each logged search query. The raw query log set has the following attributes:

- Query text: The query text of the search query that was executed.
- Search time: The time at which this search query was executed.

- Site GUID: The GUID of the site (2) (or of the site collection) from which the search query was executed.
- Search scope: The search scopes under which this search query was executed.
- Contextual Scope: The contextual search scope under which this search query was executed.
- Results Page URL: The URI of the search results page.
- Best bets: The best bets that were clicked for the query text
- Query Server: The name of the query server in which this search query was executed.
- Clicked URL: URIs clicked in the results page
- Federated location: The name of the federated location where the search query was executed.
- Query LCID: The locale identifier of the locale in which the query was executed.
- Session: An identifier that is common to a set of logged search queries that were executed by a single user within a span of time defined by the protocol client.
- User: the user who initiated the query execution.
- Tenant: the identifier of the tenant that executed the query.

3.1.1.5 Search Security Descriptor Set

The search security descriptor set maintains security information for each item. The search security descriptor set has the following attributes:

- Identifier: Security descriptor identifier
- Type: The search security descriptor type
- Security descriptor: The search security descriptor value.

3.1.1.6 Scopes

The Scopes Data Model is specified in [\[MS-SQLPADM2\]](#) section 3.1.1.4.

3.1.1.7 Summary Query Log Set

The summary query log set maintains a date wise summary of information about each executed query. The summary query log set has the following attributes:

- Query term: The query text of the search query that was executed.
- Search Date: The date on which this search query was executed.
- Site GUID: The GUID of the site (2) (or of the site collection) from which the search query was executed.
- Search scope: The search scopes, under which this search query was executed.
- Contextual Search scope: The contextual search scope under which this search query was executed.

- Query count: Number of times the search query was executed.
- Results Page URL: The URI of the search results page.
- Zero result count: Number of times zero results were returned for the search query.
- Zero best bet count: Number of times zero best bets were returned for the query text
- Uris clicked: URIs clicked in the results page.
- Best bets: The best bets that were clicked in the results page.
- Federated location: The name of the federated location where the search query was executed.

3.1.2 Timers

None.

3.1.3 Initialization

Listening endpoints are set up on the protocol server to handle inbound TDS requests. Authentication of the TDS connection to the protocol server **MUST** occur before this protocol can be used.

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

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The **T-SQL** syntax for each Stored Procedure and **result set**, and the variables they are composed of, is defined in the [\[MSDN-TSQL-Ref\]](#) protocol. In the T-SQL syntax, the variable name is followed by the type of the variable which can optionally have a length value in brackets and can optionally have a default value indicated by an equals sign followed by the default value. Unless otherwise specified, all Stored Procedures defined in this section are located in the Content Database.

For definitional clarity, a name has been assigned to any columns in the Result Sets that do not have a defined name in their current implementation. This does not affect the operation of the Result Set, as the ordinal position of any column with no defined name is expected by the front-end Web server. Such names are designated in the text using curly braces in the form *{name}*.

3.1.5.1 proc_MSS_FetchSDs

The proc_MSS_FetchSDs stored procedure is called to retrieve the search security descriptors for a list of items.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_FetchSDs (
    @joinData          VARBINARY (MAX)
```

);

@joinData: Identifiers of the security descriptors that MUST be included in the result set if they exist in the Search Security Descriptor Set as specified in section [3.1.1.5](#). This parameter MUST be encoded in the format specified by Id Blob section [2.2.1.2](#).

Return Code Values: An integer which MUST be 0.

Result Sets: This procedure MUST return the [Search Security Descriptor Result Set](#). The result set MUST NOT have more rows than the number of identifiers present in the @joinData parameter.

3.1.5.2 proc_MSS_GetDocCountEstimate

The **proc_MSS_GetDocCountEstimate** stored procedure is called to retrieve an estimate of the number of items in the MSSDocResults table, as defined in section [2.2.5.3](#).

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetDocCountEstimate (  
    @docCount          INT OUTPUT  
);
```

@docCount: The estimate of items in the MSSDocResults table.

Return Code Values: An integer which MUST be 0

Result Sets: The stored procedure MUST NOT return any result sets.

3.1.5.3 proc_MSS_GetDuplicateHashes

The **proc_MSS_GetDuplicateHashes** stored procedure is called to retrieve the Duplicate Identifier Block defined in section [2.2.1.1](#) from the MSSDocSdIds table defined in section [2.2.5.4](#) for an item.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetDuplicateHashes (  
    @UseDocId          bit,  
    @DocId             int,  
    @PathHash          bigint  
);
```

@UseDocId: MUST be a 1 when the DocId parameter is used and the PathHash parameter is ignored otherwise MUST be 0 when the PathHash parameter is used and the Docid parameter is ignored.

@DocId: MUST be used to match against the Docid column of the MSSDocSdIds table when UseDocId is 1. MUST be ignored when the UseDocIdParameter is 0.

@PathHash: When UseDocId is 0, MUST be used to match against the lIval column of the MSSDocProps table defined in section [2.2.5.2](#) to get the **Docid** of that record, which is then matched against the DocId column of the **MSSDocSdIds** as defined in section [2.2.5.4](#). When **UseDocId** is set to 1, MUST be ignored.

Return Code Values: An integer which MUST be set to zero.

Result Sets: This stored procedure MUST return the Duplicate Identifiers Result Set.

3.1.5.3.1 Duplicate Identifiers Result Set

The Duplicate Identifiers Result Set contains a Duplicate Identifier Block defined in section [2.2.1.1](#) for an item. The result set MUST contain 0 or more rows. Each row contains one Duplicate Identifier Block value.

The T-SQL for this result set is as follows:

```
DuplicateHashes          varbinary(48) NULL
```

DuplicateHashes: The value of the DuplicateHashes field in the MSSDocSdIds table (as specified in section [2.2.5.4](#)) for an item.

3.1.5.4 proc_MSS_GetKeywordInformation

The **proc_MSS_GetKeywordInformation** stored procedure is called to retrieve the definition and best bets defined for a keyword, using the data sets specified in the Best Bets and Keywords Abstract Data Model in [\[MS-SQLPDM2\]](#) section 3.1.1.2.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetKeywordInformation (  
    @Term          nvarchar(100),  
    @ConsumerGpId nvarchar(50),  
    @Keyword       nvarchar(100) OUTPUT,  
    @Definition    nvarchar(500) OUTPUT  
);
```

@Term: The term, which MAY be a synonym, used to find the unique keyword to be used for retrieving either the definition or best bets, or both.

@ConsumerGpId: The identifier of a consumer group from which to retrieve the best bet details.

@Keyword: The unique keyword associated with the **@Term**. If no **@Definition** is associated with the keyword, then this value MUST be set to NULL.

@Definition: The definition associated with the **@Keyword**.

Return Code Values:

Value	Description
0	Default return value

Result Sets:

This procedure MUST return the [Keyword Details Result Set](#).

3.1.5.4.1 Keyword Details Result Set

The Keyword Details result set contains information about a list of best bets for the specified **@Keyword** stored in the Best Bet Set as specified in [\[MS-SQLPADM2\]](#). If best bets are available for the specified **@Keyword**:

- Each row in the result set contains information about a best bet.
- The rows are sorted in ascending order specified in the Best Bet Link Set as specified in [\[MS-SQLPADM2\]](#).

Otherwise, no best bets are available for the specified **@Keyword** and no rows MUST be returned in the result set.

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

```
Title          nvarchar(100) NOT NULL,  
Url            nvarchar(2048) NOT NULL,  
Description    nvarchar(500) NOT NULL;
```

Title: Title of the best bet.

Url: URI of the best bet.

Description: Description of the best bet.

3.1.5.5 proc_MSS_GetManagedProperties

The **proc_MSS_GetManagedProperties** stored procedure is called to list managed properties from the metadata schema (as specified in [\[MS-SQLPADM2\]](#)) which were added or modified on or after the **@Idate** time.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetManagedProperties (  
    @Idate          datetime = '2000/01/01 01:01:01',  
    @GetAll         bit = 1  
);
```

@Idate: Date used to filter managed properties from the result set. The result set MUST NOT include any managed properties which have a last modified time that is older than **@Idate**

@GetAll: Flag used to indicate whether hidden managed properties should be included in the result set. If the value of **@GetAll** is 0 then the result set MUST NOT include any managed properties which have a value of 1 in the Hidden field.

Return Code Values: An integer which MUST be 0.

Result Sets:

This stored procedure MUST return a [Managed properties result set](#).

3.1.5.5.1 Managed Properties Result Set

The Managed Properties Result Set returns the list of managed properties added or updated on or after @ldate. The result set MUST contain zero or more rows.

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

PID	int,
FriendlyName	nvarchar (64),
PropertyDescription	nvarchar (2048),
ManagedType	int,
FullTextQueryable	bit,
Retrievable	bit,
Scoped	bit,
RespectPriority	bit,
RemoveDuplicates	bit,
NoDelete	bit,
NoMap	bit,
Hidden	bit,
HasMultipleValues	bit,
NoWordBreaker	bit,
NameNormalized	bit,
IncludeInMD5	bit,
Mapped	bit,
QueryIndependentRank	bit,
UserFlags	smallint,
WordBreakerOverride	int,
Weight	numeric(15,255),
LengthNormalization	numeric(15,255),
LastModified	datetime,
ForceChangeSchemaVersion	int,
MaxIndexedStringLength	int,
MaxNonIndexedStringLength	int,
MaxRetrievalLength	int,
DecimalPlaces	int,
IsInDocProps	bit,
IsInPropertyBlobOptimizedResults	bit,
IsInFixedColumnOptimizedResults	bit,
OverrideHasMultipleValues	bit,
SuppressStringNormalizer	bit,
Pronunciation	bit,
QueryPropertyBlob	bit,
NicknameExpansion	bit,
IsQIRCustomizationAllowed	bit,
DefaultForQIR	int,
SplitStringCharacters	nvarchar (64);

PID: The unique identifier for the managed property.

FriendlyName: A string that uniquely identifies the managed property.

PropertyDescription: A description for the managed property.

ManagedType: The type of the managed property as defined in Managed Type, section [2.2.1.6](#).

FullTextQueryable: A bit which MUST be 1 if the data for the managed property is kept in the **full-text index catalog**. Otherwise, it MUST be 0.

Retrievable: A bit which MUST be 1 if the data for the managed property is kept in the metadata index. Otherwise, it MUST be 0.

Scoped: A bit which MUST be 1 if the data for the managed property is kept in the **search scope index**. Otherwise, it MUST be 0.

RespectPriority: A bit which MUST be 1 if only data with highest priority (based on **mapping order**) from the **crawled properties** mapped to this managed property is used. It MUST be 0 if values from all crawled properties mapped to this managed property are used.

RemoveDuplicates: This value MUST be 1 if duplicate entries are removed. Otherwise, MUST be 0.

NoDelete: A bit which MUST be 1 if the client MUST NOT delete the managed property. Otherwise MUST be 0.

NoMap: A bit which MUST be 1 if the client MUST NOT map additional crawled properties to the managed property. Otherwise MUST be 0.

Hidden: A bit which MUST be 1 if the managed property is intended to be hidden from display during administrative operations. Otherwise, it MUST be 0.

HasMultipleValues: A bit which MUST be 1 if the value of the managed property is configured to contain multiple values. Otherwise, it MUST be 0. This configuration MUST NOT be ignored if the value of the **OverrideHasMultipleValues** property is 0.

NoWordBreaker: This parameter MUST be ignored by the client.

NameNormalized: A bit which MUST be 0 if the values of this managed property are not to be normalized by the **index server**. Otherwise, it MUST be 1.

IncludeInMD5: A bit which MUST be 1 if values mapped to this managed property are used to determine if the item has changed. Otherwise, it MUST be 0.

Mapped: A bit which MUST be 1 when the managed property is a URL that is manipulated by **alternate access mappings**. Otherwise, it MUST be 0.

QueryIndependentRank: A bit which MUST be 1 when the managed property participates in **query independent rank**. Otherwise, it MUST be 0.

UserFlags: A user-specified numeric value associated with the managed property.

WordBreakerOverride: This parameter MUST be ignored by the client.

Weight: A decimal value used to adjust **property oriented rank**.

LengthNormalization: A decimal value used to adjust property oriented rank.

LastModified: The date and time of the last change to the managed property.

ForceChangeSchemaVersion: This parameter MUST be ignored by the client.

MaxIndexedStringLength: If the **ManagedType** parameter is not equal to 1 and the **IsInDocProps** parameter is not equal to 1 then the protocol client MUST ignore this parameter. Otherwise, this parameter contains the maximum number of characters for this managed property that the protocol client MUST store in or read from the strVal column of the MSSDocProps table, as specified in section [2.2.5.2](#).

MaxNonIndexedStringLength: If the **ManagedType** parameter is not equal to 1 and the **IsInDocProps** parameter is not equal to 1 then the protocol client MUST ignore this parameter. Otherwise, this parameter contains the maximum number of characters for this managed property that the protocol client MUST store in or read from the strVal2 column of the MSSDocProps table, as specified in section [2.2.5.2](#).

MaxRetrievalLength: If the **ManagedType** parameter is not equal to 1 and the **IsInFixedColumnOptimizedResults** parameter is not equal to 1 then the protocol client MUST ignore this parameter. Otherwise, this parameter contains the maximum number of characters for this managed property that the protocol client MUST store in or read from the column of the MSSDocResults table, as specified in section [2.2.5.3](#).

DecimalPlaces: If the **ManagedType** parameter is not equal to 3 then the protocol client MUST ignore this parameter. Otherwise, this parameter contains the number of decimal places for this managed property which are stored in the value of the lVal column of the MSSDocProps table, as specified in section [2.2.5.2](#).

IsInDocProps: A flag indicating the storage location for this managed property. If the value of this parameter is 0 then the managed property value MUST be stored in the MSSDocResults table as specified in section [2.2.5.3](#), otherwise it MUST be stored in the MSSDocProps table, as specified in section [2.2.5.2](#).

IsInPropertyBlobOptimizedResults: A flag indicating the storage location for this managed property within the MSSDocResults table, as specified in section [2.2.5.3](#). If the value of this parameter is 0 then the managed property value MUST NOT be stored in the PropertyBlob column of the MSSDocResults table. If the value of this parameter is 1 then the managed property value MUST be stored in the PropertyBlob column of the MSSDocResults table. If the value of the **IsInDocProps** parameter is 1 then the protocol client MUST ignore the value of this parameter.

IsInFixedColumnOptimizedResults: A flag indicating the storage location for this managed property within the MSSDocResults table, as specified in section [2.2.5.3](#). If the value of this parameter is 0 then the managed property value MUST NOT be stored in one of the column of the MSSDocResults table other than the DocId and PropertyBlob columns. If the value of this parameter is 1 then the managed property value MUST be stored in one of columns of the MSSDocResults table other than the DocId or PropertyBlob columns. If the value of the **IsInDocProps** parameter is 1 then the protocol client MUST ignore the value of this parameter.

OverrideHasMultipleValues: A flag indicating whether the **HasMultipleValues** parameter is the sole mechanism for the protocol client to use for determining whether the managed property is multi-valued. If the value of the **OverrideHasMultipleValues** parameter is 0 then the protocol client MUST NOT ignore the value of the **HasMultipleValues** parameter, otherwise the protocol client MUST use logic to determine whether the managed property is multi-valued.

SuppressStringNormalizer: If the value of this parameter is 1 then the protocol client MUST NOT perform string normalization on the managed property value. Otherwise, the protocol client MUST allow string normalization to be performed on the managed property value.

Pronunciation: If the value of this parameter is 0 then the protocol client MUST not perform phonetic analysis of the managed property value. Otherwise, the protocol client MUST perform phonetic analysis to be performed on the managed property value.

QueryPropertyBlob: MUST contain the same value as the **IsInPropertyBlobOptimizedResults** parameter.

NicknameExpansion: This value MUST be set to 1 if the managed property is compared to the nickname mappings at query time. Otherwise, it MUST be 0.

IsQIRCustomizationAllowed: If the value of this parameter is 0 then the protocol client MUST NOT modify the static ranking default value for this managed property as specified in [MS-SQLPADM2]. Otherwise, the protocol client MUST allow customization of the static ranking default value on the managed property value.

DefaultForQIR: The value of this parameter MUST contain the static ranking default value for this managed property as specified in [MS-SQLPADM2].

The **GetManagedProperties** result set returns the list of managed properties which were added or updated, on or after the *@ldate*. The result set MUST contain zero or more rows.

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

PID	int,
FriendlyName	nvarchar (64) ,
PropertyDescription	nvarchar (2048) ,
ManagedType	int,
FullTextQueryable	bit,
Retrievable	bit,
Scoped	bit,
RespectPriority	bit,
RemoveDuplicates	bit,
NoDelete	bit,
NoMap	bit,
Hidden	bit,
HasMultipleValues	bit,
NoWordBreaker	bit,
NameNormalized	bit,
IncludeInMD5	bit,
Mapped	bit,
QueryIndependentRank	bit,
UserFlags	smallint,
WordBreakerOverride	int,
Weight	float,
LengthNormalization	float,
LastModified	datetime,
ForceChangeSchemaVersion	int,
MaxIndexedStringLength	int,
MaxNonIndexedStringLength	int,
MaxRetrievalLength	int,
DecimalPlaces	int,
IsInDocProps	bit,
IsInPropertyBlobOptimizedResults	bit,
IsInFixedColumnOptimizedResults	bit,
OverrideHasMultipleValues	bit,
SuppressStringNormalizer	bit,
Pronunciation	bit,
QueryPropertyBlob	bit,
NicknameExpansion	bit,
IsQIRCustomizationAllowed	bit,
DefaultForQIR	int,
SplitStringCharacters	nvarchar (64) ;

PID: The unique identifier for the managed property. This value MUST NOT be NULL.

FriendlyName: A string that uniquely identifies the managed property. This value MUST NOT be NULL.

PropertyDescription: The description of the managed property.

ManagedType: The type, as specified in Section [2.2.6.2](#), of the managed property.

FullTextQueryable: This value MUST be 1 if the data for the managed property is kept in the full-text index catalog. Otherwise, it MUST be 0.

Retrievable: This value MUST be 1 if the data for the managed property is kept in the metadata index. Otherwise, it MUST be 0.

Scoped: This value MUST be 1 if the data for the managed property is kept in the search scope index. Otherwise, it MUST be 0.

RespectPriority: This value MUST be 1 if only data from the crawled property mapped to this managed property with highest priority of its mapping order is used. It MUST be 0 if values from all crawled properties mapped to this managed property are used.

RemoveDuplicates: This value MUST be 1 if the duplicate entries are removed. Otherwise, MUST be 0.

NoDelete: This value MUST be 1 if this property can never be deleted from the **metadata schema**. Otherwise, it MUST be 0.

NoMap: This value MUST be 1 if this property can never have its **search property mappings** altered. Otherwise, it MUST be 0.

Hidden: This value MUST be set to 1 to specify that the property is an internal property. Otherwise, it MUST be set to 0.

HasMultipleValues: This value MUST be 1 if the values of the managed property can contain multiple values. Otherwise, it MUST be 0.

NoWordBreaker: This parameter MUST be ignored by the client.

NameNormalized: This value MUST be 1 if the values of this managed property are to be normalized by the index server. Otherwise, it MUST be 0.

IncludeInMDS: This value MUST be 1 if values mapped to this managed property are used to determine if the item has changed. Otherwise, it MUST be 0.

Mapped: This value MUST be 1 when the property is a **URL** that subject to alternate access mappings. Otherwise it MUST be 0.

QueryIndependentRank: This value MUST be 1 when the property participates in query independent rank. Otherwise, it MUST be 0.

UserFlags: The flag that can be retrieved and set by an administrator that is open to custom applications that use the public schema **object model** to get and set these. This value MUST NOT be NULL.

WordBreakerOverride: This parameter MUST be ignored by the client.

Weight: A decimal value used to adjust property oriented rank. This value MUST NOT be NULL.

LengthNormalization: A decimal value used to adjust property oriented rank. This value MUST NOT be NULL.

LastModified: The date and time of the last change to the managed property. This value MUST NOT be NULL.

ForceChangeSchemaVersion: This parameter MUST be ignored by the client.

MaxIndexedStringLength: The maximum number of characters persisted in the string value column in the *MSSDocProps* table defined in [MS-SQLPQ2] section 2.2.5.1. If the string value length is greater than *MaxIndexedStringLength*, the string is truncated to *MaxIndexedStringLength*. This value MUST NOT be NULL.

MaxNonIndexedStringLength: If the string value length in the *MSSDocProps* table, defined in [MS-SQLPQ2] section 2.2.5.1, is greater than the size allowed by the *MaxIndexedStringLength*, the string is truncated to *MaxIndexedStringLength* and the string overflow is stored in the *strVal2* column, in the *MSSDocProps* table defined in [MS-SQLPQ2] section 2.2.5.1, up to the size allowed by the *MaxNonIndexedStringLength*. This value MUST NOT be NULL.

MaxRetrievalLength: The maximum number of characters persisted for a fixed-length string property in the *MSSDocResults* table defined in [MS-SQLPQ2] section 2.2.5.2. This MUST NOT be NULL.

DecimalPlaces: The number of floating point decimal places that must be honored in the metadata index. This value MUST NOT be NULL.

IsInDocProps: This value MUST be 1 if the managed property is stored in the metadata index. Otherwise, MUST be 0.

IsInPropertyBlobOptimizedResults: This value MUST be 1 if the managed property is persisted in the *PropertyBlob* column of the *MSSDocResults* table defined in [MS-SQLPQ2] section 2.2.5.2. Otherwise, MUST be 0.

IsInFixedColumnOptimizedResults: This value MUST be 1 if the managed property is persisted in the *MSSDocResults* table defined in [MS-SQLPQ2] section 2.2.5.2. Otherwise, MUST be 0.

OverrideHasMultipleValues: This value MUST be 1 if the metadata schema object model MUST NOT change the *HasMultipleValues* parameter because it has been set explicitly. Otherwise, MUST be 0.

SuppressStringNormalizer: This value MUST be 1 if the string normalization for the managed property is to be skipped. Otherwise this value MUST be 0.

Pronunciation: This value MUST be 1 if the managed property needs a pronunciation string. Otherwise, MUST be 0.

QueryPropertyBlob: This value MUST be 1 if the managed property is persisted in the *PropertyBlob* column of the *MSSDocResults* table defined in [MS-SQLPQ2] section 2.2.5.2. Otherwise, MUST be 0.

NicknameExpansion: This value MUST be set to 1 if the managed property is compared to the nickname mappings at query time. Otherwise, it MUST be 0.

IsQIRCustomizationAllowed: This value MUST be 1 if the managed property supports query independent rank and customization of query independent rank is allowed. Otherwise, MUST be 0.

DefaultForQIR: The default value for query independent rank. If the managed property does not participate in query independent rank then this value MUST be 0.

SplitStringCharacters: This value MUST contain characters which are used to split the string data for the managed property into separate strings which do not contain the SplitStringCharacters. Otherwise, it MUST be NULL.

3.1.5.6 `proc_MSS_GetManagedPropertyAliases`

The `proc_MSS_GetManagedPropertyAliases` stored procedure is called to list the aliases for managed properties from the metadata schema which were added or modified on or after the `@LastModified` time.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetManagedPropertyAliases (  
    @LastModified          datetime  
) ;
```

@LastModified: Date used to filter managed properties from the result set. The result set MUST NOT include any managed properties which have a last modified time that is older than **@LastModified**.

Return Code Values: An integer which MUST be 0.

Result Sets:

This stored procedure MUST return the [Managed Property Aliases Result Set](#).

3.1.5.6.1 **Managed Property Aliases Result Set**

The Managed Property Aliases Result Set returns the list of aliases for managed properties added or updated on or after **@LastModified** time. The result set MUST contain zero or more rows.

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

```
PID                int,  
alias              nvarchar(2048),  
LastModified       datetime;
```

PID: The unique identifier of a managed property.

alias: An alternate string name which identifies a managed property.

LastModified: This parameter MUST be ignored by the client.

3.1.5.7 `proc_MSS_GetMultipleResults`

The `proc_MSS_GetMultipleResults` stored procedure is called to retrieve one or more of the following:

- Extracted definitions for the specified term.
- High confidence results for the specified term or author.
- Managed properties for a list of items.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_MSS_GetMultipleResults (
    @RequestTypes      int,
    @Term              nvarchar(100),
    @Author            nvarchar(100),
    @PropertiesSelectSql nvarchar(2048),
    @joinData          varbinary(max)
);

```

@RequestTypes: The type of result expected. Multiple values can be combined as a bitmask to indicate that multiple result sets MUST be returned. This parameter MUST contain at least one of the following bitmask values:

Value	Description
0x00000001	Managed properties for a list of items MUST be returned in a Search common property result set and a Search property result set .
0x00000004	The High confidence result set for @Term MUST be returned.
0x80000000	The Definitions result set for @Term MUST be returned.

@Term: The term for which the extracted definition or high confidence results will be retrieved.

@Author: The restrictions imposed on author managed property.

@PropertiesSelectSql: The T-SQL statement which contains a list of managed properties that, when used with the blob specified in **@joinData**, returns the requested list of managed properties for the list of items in the blob. The T-SQL statement MUST retrieve data from the MSSDocProps table, as defined in section [2.2.5.2](#) and the MSSDocResults table, as defined in section [2.2.5.3](#).

@joinData: A list of item identifiers and their rank contained in an Id Value Pairs Blob as specified in section [2.2.3.2](#).

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result sets in the following order:

- MUST return the Definitions result set if the **@RequestTypes** parameter indicates that it is required. Otherwise MUST NOT return Definitions result set.
- MUST return the High confidence result set if the **@RequestTypes** parameter indicates that it is required. Otherwise MUST NOT return High confidence result set.
- MUST return the Search common property result set and the Search property result set if the **@RequestTypes** parameter indicates that they are required. Otherwise MUST NOT return the Search common property result set or the Search property result set.

3.1.5.7.1 Definitions Result Set

The Definitions result set contains information about a list of items that contains the definition for the **@Term**. Each row in the result set contains the identifier of an item and the corresponding definition related information. The result set MUST contain 10 or fewer rows.

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

```

DocId          int,

```

```

Sdid          int,
Sentence      nvarchar(255),
TermOffset    int,
TermLength    int,
{href}        nvarchar(2450),
{title}       nvarchar(2450);

```

DocId: The identifier of an item.

Sdid: The search security descriptor identifier of the item.

Sentence: The extracted definition of the item.

TermOffset: The character offset at which the specified term appears in **Sentence**.

TermLength: The number of characters in the specified term.

{href}: The URI of the item.

{title}: The title of the item.

3.1.5.7.2 High Confidence Result Set

The High confidence result set contains information about a list of items for which at least one **high confidence property** is equal to **@Term** or whose author managed property is equal to **@Author**. The result set contains zero or more rows per item. The result set contains zero rows if there are no high confidence results for the specified **@Term**.

Each row in the result set contains an item identifier, property identifier and the corresponding managed property value for the item. The managed property value is the concatenation of **strVal** and **strVal2**.

The T-SQL for this result set is as follows:

```

Docid          int,
pid            int,
strVal         nvarchar(450),
strVal2       nvarchar(max);

```

DocId: The identifier of an item.

pid: The identifier of a high confidence property.

strVal: The first part of the value of the high confidence property.

strVal2: The second part of the value of the high confidence property.

3.1.5.7.3 Search Common Property Result Set

The Search common property result set contains item identifiers and managed property values for a list of items stored in **@joinData**. The result set MUST contain one row per item. For each item, the result set MUST contain all existing values for all managed properties which exist in the MSSDocResults table as defined in section [2.2.5.3](#). The T-SQL for this result set is as follows:

```

DocId          int,

```

SummaryBlobSize	int,
Size	bigint,
LastModified	bigint,
IsDocument	bit,
IsPictureUrl	bit,
Author	nvarchar(4000),
Title	nvarchar(4000),
Url	nvarchar(4000),
PictureThumbnailUrl	nvarchar(4000),
ContentClass	nvarchar(4000),
FileExtension	nvarchar(4000),
Tags	nvarchar(4000),
PropertyBlob	varbinary(max),
{PropertyBlobLength}	bigint,
PopularSocialTags	nvarchar(4000),
SiteName	nvarchar(4000),
Description	nvarchar(4000),
ParentLink	nvarchar(4000),
NumberOfMembers	int,
PictureHeightAndWidth	bigint,
DisplayDate	bigint;

DocId: The identifier of the item.

SummaryBlobSize: The value of the SummaryBlobSize field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

Size: The value of the Size field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

LastModified: The value of the LastModified field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

IsDocument: The value of the IsDocument field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

IsPictureUrl: The value of IsPictureUrl field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

Author: The value of the Author field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

Title: The value of the Title field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

Url: The value of the Url field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

PictureThumbnailUrl: The value of the PictureThumbnailUrl field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

ContentClass: The value of the ContentClass field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

FileExtension: The value of the FileExtension field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

Tags: The value of the Tags field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

PropertyBlob: The value of the PropertyBlob field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

{PropertyBlobLength}: The length of the **PropertyBlob** field value in bytes. If the PropertyBlob field value is NULL then the value of {PropertyBlobLength} MUST be 0.

PopularSocialTags: The value of the PopularSocialTags field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

SiteName: The value of the SiteName field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

Description: The value of the Description field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

ParentLink: The value of the ParentLink field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

NumberOfMembers: The value of the NumberOfMembers field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

PictureHeightAndWidth: The value of the PictureHeightAndWidth field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

DisplayDate: The value of the DisplayDate field in the MSSDocResults table (as specified in section [2.2.5.3](#)) for the item.

3.1.5.7.4 Search Property Result Set

The Search property result set contains item identifiers and managed property values for a list of items stored in **@joinData**. The result set MUST contain zero or more rows per item. For each item, the result set MUST contain all existing values for all managed properties in **@PropertiesSelectSql** which exist in the MSSDocProps table (as specified in section [2.2.5.2](#)). Each row in the result set contains an item identifier, a property identifier and corresponding value of the managed property.

In each row, the value of the managed property MUST be contained in the column that matches the variant type of the managed property. See section [2.2.5.2](#) for more details.

The T-SQL for this result set is as follows:

```
Docid          int,
pid            int,
llVal          bigint,
strVal         nvarchar(450),
strVal2        nvarchar(max),
{strVal2Length} bigint;
```

DocId: The identifier of the item.

pid: The identifier of a managed property associated with the item.

llVal: The value of the llVal field in the MSSDocProps table (as specified in section [2.2.5.2](#)) for the item.

strVal: The value of the strVal field in the MSSDocProps table (as specified in section [2.2.5.2](#)) for the item.

strVal2: The value of the strVal2 field in the MSSDocProps table (as specified in section [2.2.5.2](#)) for the item.

{strVal2Length}: The length of the strVal2 field in bytes. It MUST be 0 if strVal2 is NULL.

3.1.5.8 proc_MSS_GetPronunciations

The **proc_MSS_GetPronunciations** stored procedure is called to retrieve all values from the strVal field of the MSSDocProps table (as specified in section [2.2.5.2](#)) for the specified managed property identifier.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetPronunciations (  
    @Pid                int);
```

@pid: The managed property identifier for which to return all strval field values.

Return Code Values: An integer which MUST be 0.

Result Sets: This stored procedure MUST return the Pronunciations Result Set.

3.1.5.8.1 Pronunciations Result Set

The Pronunciations result set contains all string values for a managed property. The result set MUST contain 0 or more rows. Each row contains one string value.

The T-SQL for this result set is as follows:

```
strVal                nvarchar(450);
```

strVal: The value of the strVal field in the MSSDocProps table (as specified in section [2.2.5.2](#)) for a managed property value.

3.1.5.9 proc_MSS_GetQuerySuggestions

The **proc_MSS_GetQuerySuggestions** stored procedure is called to retrieve potential query text given partial query text entered by a user.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetQuerySuggestions (  
    @q                nvarchar(1024),  
    @qHash            int,  
    @nTerms           int,  
    @nMaxSuggestions  int,  
    @LastTerm         nvarchar(256),  
    @qHash0           int,  
    @qHash1           int,  
    @qHash2           int,  
    @qHash3           int,  
    @qHash4           int,
```

```

    @qHash01          int,
    @qHash12          int,
    @qHash23          int,
    @qHash34          int
);

```

@q: The partial query text for which to retrieve query text suggestions.

@qHash: MUST contain the result of a **cyclic redundancy check (CRC)** calculation of **@q**.

@nTerms: MUST contain the number of tokens in **@q**.

@nMaxSuggestions: MUST contain the maximum number of suggestions to be returned in the [Query suggestions result set](#).

@LastTerm: MUST contain the last token in **@q** or an empty string if **@q** is empty.

@qHash0: Identifier of first token to use for retrieving query suggestions. It MUST contain the result of a CRC calculation of the first token in **@q**. MUST be 0 if there are no tokens in **@q**.

@qHash1: Identifier of second token to use for retrieving query suggestions. It MUST contain the result of a CRC calculation of the second token in **@q**. MUST be 0 if there are fewer than 2 tokens in **@q**.

@qHash2: Identifier of third token to use for retrieving query suggestions. It MUST contain the result of a CRC calculation of the third token in **@q**. MUST be 0 if there are fewer than 3 tokens in **@q**.

@qHash3: Identifier of fourth token to use for retrieving query suggestions. It MUST contain the result of a CRC calculation of the fourth token in **@q**. MUST be 0 if there are fewer than 4 tokens in **@q**.

@qHash4: Identifier of fifth token to use for retrieving query suggestions. It MUST contain the result of a CRC calculation of the fifth token in **@q**. MUST be 0 if there are fewer than 5 tokens in **@q**.

@qHash01: MUST contain the CRC of the first token concatenated with the second token.

@qHash12: MUST contain the CRC of the second token concatenated with the third token.

@qHash23: MUST contain the CRC of the third token concatenated with the fourth token.

@qHash34: MUST contain the CRC of the fourth token concatenated with the fifth token.

Return Code Values: An integer which MUST be 0.

Result Sets:

This stored procedure MUST return the Query suggestions result set.

3.1.5.9.1 Query Suggestions Result Set

The Query suggestions result set contains query text suggestions. The result set MUST contain at most **@nMaxSuggestions** rows. Each row contains information about one query text suggestion.

The T-SQL for this result set is as follows:

```

queryString      nvarchar(1024),
tcnt             int,
termToReplace    nvarchar(1000),
acronymToMatch   nvarchar(50),
queryCount       int,
mtcnt           int,
xcnt            int;

```

queryString: A string containing the query text suggestion.

tcnt: The count of occurrences of the tokens in **@q** that are present in **queryString**.

termToReplace: The text to use when replacing the original query text.

acronymToMatch: The token that MUST be replaced in the original query text.

queryCount: The number of queries which have been issued with the suggested query text.

mtcnt: The number of concatenated terms in the original query text that have been found in the new query text.

xcnt: The number of terms in the original query text that have been found in the new query text.

3.1.5.10 proc_MSS_GetRankingModels

The `proc_MSS_GetRankingModels` stored procedure is specified in [\[MS-SQLPADM2\]](#) section 3.1.5.100.

3.1.5.11 proc_MSS_GetSchemaHighLevelInfo

The `proc_MSS_GetSchemaHighLevelInfo` stored procedure is called to retrieve last modified and last deleted timestamps from the metadata schema.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetSchemaHighLevelInfo ();
```

Return Code Values: An integer which MUST be 0.

Result Sets: This stored procedure MUST return the [Schema High Level Info Result Set](#).

3.1.5.11.1 Schema High Level Info Result Set

The `GetSchemaHighLevelInfo` result set returns global datetime timestamps reflecting changes made to the metadata schema. The result set MUST contain zero rows or one row.

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

```

LastCatChange      datetime,
LastCPDelete       datetime,
LastCPAddsBenignModified  datetime,
LastURIAdds        datetime,
LastURIModifiedDeleted  datetime,
LastManagedProp   datetime,
LastGlobalProps    datetime,

```


LastManagedPropDeleted	datetime,
LastSmpDelete	datetime,
LastAliasAdd	datetime,
LastAliasOther	datetime,
ForceChangeSchemaVersion	int;

LastCatChange: a timestamp that contains the last local time of the server when a **crawled property category** was added, modified, or deleted.

LastCPDelete: a timestamp that contains the last local time of the server when a crawled property was deleted.

LastCPAddsBenignModified: a timestamp that contains the last local time of the server when a crawled property is added or modified or when a mapping from a crawled property to a managed property is added, changed or deleted.

LastURIAdds: This parameter MUST be ignored by the client.

LastURIModifiedDeleted: This parameter MUST be ignored by the client.

LastManagedProp: a timestamp that contains the last local time of the server when a managed property is added or deleted.

LastGlobalProps: a timestamp that contains the last local time of the server when a schema parameter is added or modified.

LastManagedPropDeleted: a timestamp that contains the last local time of the server when a crawled property or managed property is deleted.

LastSmpDelete: This parameter MUST be ignored by the client.

LastAliasAdd: This parameter MUST be ignored by the client.

LastAliasOther: This parameter MUST be ignored by the client.

ForceChangeSchemaVersion: This parameter MUST be ignored by the client.

3.1.5.12 `proc_MSS_GetSchemaParameters`

The `proc_MSS_GetSchemaParameters` stored procedure is called to retrieve a list of parameters from the metadata schema.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetSchemaParameters ();
```

Return Code Values: An integer which MUST be 0.

Result Sets:

This stored procedure MUST return a [Schema parameters result set](#).

3.1.5.12.1 Schema Parameters Result Set

The Schema parameters result set returns the list of parameters from the metadata schema. The result set MUST contain zero or more rows, each corresponding to a single parameter.

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

```
ParamName          nvarchar(40),
IsString           bit,
strValue           nvarchar(256),
fltValue           numeric(15,255);
```

ParamName: The name of the schema parameter.

IsString: If set to 1, the strValue field MUST be set to the string value of the schema parameter. Otherwise, it MUST be set to 0 and the value of the schema parameter MUST be returned in the fltValue field.

strValue: The string value of the parameter. This field MUST be ignored when IsString is set to 0

fltValue: The floating-point value of the parameter. This field MUST be ignored when IsString is set to 1.

3.1.5.13 proc_MSS_GetStaticRankingFeatures

The proc_MSS_GetStaticRankingFeatures stored procedure is specified in [\[MS-SQLPADM2\]](#) section 3.1.5.127.

3.1.5.14 proc_MSS_GetVectorI4ConfigurationProperty

The **proc_MSS_GetVectorI4ConfigurationProperty** stored procedure is called to retrieve the integer values of a **vector configuration property**. The vector configuration property value MUST be of integer type.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_GetVectorI4ConfigurationProperty (
    @Name          nvarchar(300)
);
```

@Name: The name of the vector configuration property.

Return Code Values: This stored procedure returns an integer value that MUST be ignored.

Result Sets: The stored procedure MUST return the [VectorI4ConfigurationProperty](#) result set.

3.1.5.14.1 VectorI4ConfigurationProperty Result Set

The GetVectorI4ConfigurationProperty result set returns a list of the property values. The result set MUST contain zero or more rows, each corresponding to a value of the specified vector configuration property.

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

```
{VectorValue}          int;
```

{VectorValue}: The value of the property.

3.1.5.15 proc_MSS_PrefetchSDs

The proc_MSS_PrefetchSDs stored procedure is called to retrieve a range of search security descriptors.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_PrefetchSDs (  
    @firstSdid          INT,  
    @rowsToGet         INT  
);
```

@firstSdid: Identifier of the first security descriptor that MUST be included in the result set if it exists in the Search Security Descriptor Set as specified in section [3.1.1.5](#).

@rowsToGet: The maximum number of rows that MUST be included in the result set.

Return Code Values: An integer which MUST be 0.

Result Sets: The stored procedure MUST return the [Search Security Descriptor Result Set](#). The result set MUST NOT contain more rows than @rowsToGet. The result set MUST NOT contain any items which have a security descriptor identifier which is less than @firstSdid.

3.1.5.16 proc_MSS_QLog_AllocateClickIds

The proc_MSS_QLog_AllocateClickIds stored procedure is called to reserve a list of click identifiers that can be used to uniquely log search query information. The protocol client MUST log search query information using the click identifiers returned in the [Click Id Result Set](#).

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_QLog_AllocateClickIds (  
    @numIds            int  
);
```

@numIds: The number of consecutive numbers that this procedure MUST reserve to the search query log, beginning with the value returned in the result set when this stored procedure is executed.

Return Code Values:

Value	Description
0	Default return value

Results Sets:

This procedure MUST NOT return any result set if @numIds is NULL. Otherwise, this procedure MUST return the Click Id Result Set.

3.1.5.16.1 Click Id Result Set

The click id result set returns a 64-bit number that MUST be used as the click identifier for any new items stored in the [MSSQLLogNonClickedUnprocessed](#) and [MSSQLLogUnprocessed](#) tables. The result set MUST contain one row.

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

```
nextId          bigint;
```

nextId: The first click identifier that the calling protocol client can use to uniquely log search query information.

The protocol client MUST use only number(s) from **nextId** to **nextId+@numIds-1** to uniquely log search query information. The `proc_MSS_QLog_AllocateClickIds` stored procedure MUST NOT return a value that is less than **nextId+@numIds-1** from any of its subsequent invocations.

3.1.5.17 proc_MSS_QLog_InsertQueryInfo

The `proc_MSS_QLog_InsertQueryInfo` stored procedure is called to populate the [Historical Query Log Set](#), [Detailed Query Log Set](#) and [Summary Query Log Set](#) from the [Raw Query Log Set](#) data in the [MSSQLLogUnprocessed](#) and [MSSQLLogNonClickedUnprocessed](#) tables. This stored procedure MUST truncate all the information from `MSSQLLogUnprocessed` and `MSSQLLogNonClickedUnprocessed` tables after processing the information.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_QLog_InsertQueryInfo ();
```

Return Code Values: An integer which MUST be 0.

Result Sets: SHOULD NOT return any result sets. The protocol client MUST ignore any result sets returned by this stored procedure.

3.1.5.18 proc_MSS_ScopeCacheGetChangedConsumers

The `proc_MSS_ScopeCacheGetChangedConsumers` stored procedure is called to retrieve the names of all search scope consumers who own search scope display groups, search scopes or search scope rules that have been changed since the specified version.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_ScopeCacheGetChangedConsumers (
    @Since          bigint
);
```

@Since: A version whose value MUST be a [LastUpdate data type](#).

Return values:

Value	Description
0	Default return value

Result Sets:

This procedure MUST return the [Changed Consumers Result Set](#) and the [Last Consumer Change Identifier Result Set](#).

3.1.5.18.1 Changed Consumers Result Set

The Changed consumers result set contains information about the names of the search scope consumers who own search scope display groups, search scopes or search scope rules that have been changed since the specified version. The result set MUST contain one row for each single search scope consumer that has changed since the **@Since** version.

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

```
ConsumerName          nvarchar(60) NOT NULL;
```

ConsumerName: The name of the search scope consumer.

3.1.5.18.2 Last Consumer Change Identifier Result Set

The Last consumer change identifier result set contains information about the version for the specified search scope consumer. The result set MUST contain one row.

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

```
LastConsumerChangeID  int NOT NULL;
```

LastConsumerChangeID: The version whose value MUST be a [LastConsumerChangeID](#).

3.1.5.19 proc_MSS_ScopeCacheGetChanges

The `proc_MSS_ScopeCacheGetChanges` stored procedure is called to retrieve search scopes, search scope display groups, and membership information for the specified search scope consumer.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MSS_ScopeCacheGetChanges (  
    @ConsumerName          nvarchar(60)  
);
```

@ConsumerName: The name of the search scope consumer.

Return Code Values:

Value	Description
0	Default return value

Result Sets:

This procedure MUST return one each of [Scopes Result Set](#), [Display Groups Result Set](#), [Compiled Scopes Result Set](#) and [Last Update Result Set](#) in that order.

3.1.5.19.1 Scopes Result Set

The Scopes result set contains information about all scopes for the specified search scope consumer. The result set MUST contain zero or more rows, each corresponding to a single scope.

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

```
ScopeID          int NOT NULL,  
Name             nvarchar(60) NOT NULL,  
Description      nvarchar(300) NOT NULL,  
AlternateResultsPageUrl nvarchar(2048) NOT NULL,  
CompilationState smallint NOT NULL,  
Filter           nvarchar(2047) NOT NULL;
```

ScopeID: Unique identifier of the search scope.

Name: The name of the search scope.

Description: The description of the search scope.

AlternateResultsPageUrl: The URI of an alternate search results page to display the results of a search performed on this search scope. This value MUST be set to NULL if no alternate search results page is defined.

CompilationState: The **search scope compilation** state of the given search scope. The value MUST be a Compilation Type as specified in [\[MS-SQLPDM2\]](#) section 2.2.1.5.

Filter: Reserved. This parameter MUST be ignored by the client.

3.1.5.19.2 Display Groups Result Set

The Display groups result set contains information about display groups for the specified search scope consumer. The result set MUST contain zero or more rows, each corresponding to a single search scope display group.

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

```
DisplayGroupID   int NOT NULL,  
Name             nvarchar(60) NOT NULL,  
DefaultScopeID  int NOT NULL;
```

DisplayGroupID: The unique identifier of the search scope display group.

Name: The name of the search scope display group.

DefaultScopeID: The unique identifier of the **default search scope** of the search scope display group.

3.1.5.19.3 Compiled Scopes Result Set

The Compiled scopes result set contains information about **compiled search scopes** sorted in ascending order by their **Rank**. The result set MUST contain zero or more rows, each corresponding to a single compiled search scope.

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

```
DisplayGroupID      int NOT NULL,  
ScopeID             int NOT NULL,  
Rank                int NOT NULL;
```

DisplayGroupID: The unique identifier of the search scope display group of the compiled search scope.

ScopeID: The unique identifier of the compiled search scope.

Rank: This contains a unique ordinal position of this search scope display group for display purposes.

3.1.5.19.4 Last Update Result Set

The Last update result set contains information about the last update timestamp value for the specified search scope consumer. The result set MUST contain zero or one row.

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

```
LastUpdate          bigint NOT NULL;
```

LastUpdate: The last update value of the search scope consumer. The value MUST be a LastUpdate value as specified in [\[MS-SQLPADM2\]](#), section [3.1.1.4](#).

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Microsoft SharePoint Server 2010 Client Details

The protocol client role that is served by the search query and log server role is described in this section. The client role can make requests for query execution, managed property value retrieval operations, and search query logging operations.

An external client (such as a user) can also send a request to the protocol client to log a search query. In this case, the protocol client calls a stored procedure in the server to perform the requested operation.

3.2.1 Abstract Data Model

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

The protocol client maintains the following objects for this protocol:

- Query Result Candidate

- Result Document Property

The following subsections define each of these data structures.

3.2.1.1 Query Result Candidate Document Set

The query result candidate document set is used to store a list of potential query results during search query execution. As the search query is executed by the protocol client, the query result candidate set contains a working list of items which are further reduced as the search query is processed. The query result candidate set has the following attribute:

- Item identifier

3.2.1.2 Result Document Property Set

The result document set contains a list of item that is part of a search result set with zero or more properties. The attributes of an item property set are:

- Item identifier
- Property identifier
- Property value

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

A user sends a request to the protocol client to perform a search query. In this case, the protocol client performs a list of tasks necessary to return items which match this search query. The tasks include retrieving items from the protocol server, eliminating certain items based on security information or because of duplicate results, and retrieving managed properties for a list of items using the following sequence:

1. The task of retrieving item properties from the protocol server **MUST** be accomplished by querying the MSSDocProps (as defined in section [2.2.5.2](#)) or MSSDocSdids (as defined in section [2.2.5.4](#)) tables.
2. The item elimination tasks **MUST** be accomplished by executing the proc_MSS_FetchSDs (as specified in section [3.1.5.1](#)) and proc_MSS_GetDuplicateHashes (as specified in section [3.1.5.15](#)) stored procedures.
3. The task of retrieving item property values from the protocol server **MUST** be accomplished by calling proc_MSS_GetMultipleResults (as specified in section [3.1.5.7](#)) to retrieve properties for the query results.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Windows SharePoint Services 4.0 Server Details

The search query server role is described in this section. This role serves requests for search query execution and managed properties retrieval operations.

3.3.1 Abstract Data Model

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

The protocol server maintains the following set of data for this protocol within the metadata schema. Data within the protocol server are maintained until updated or removed.

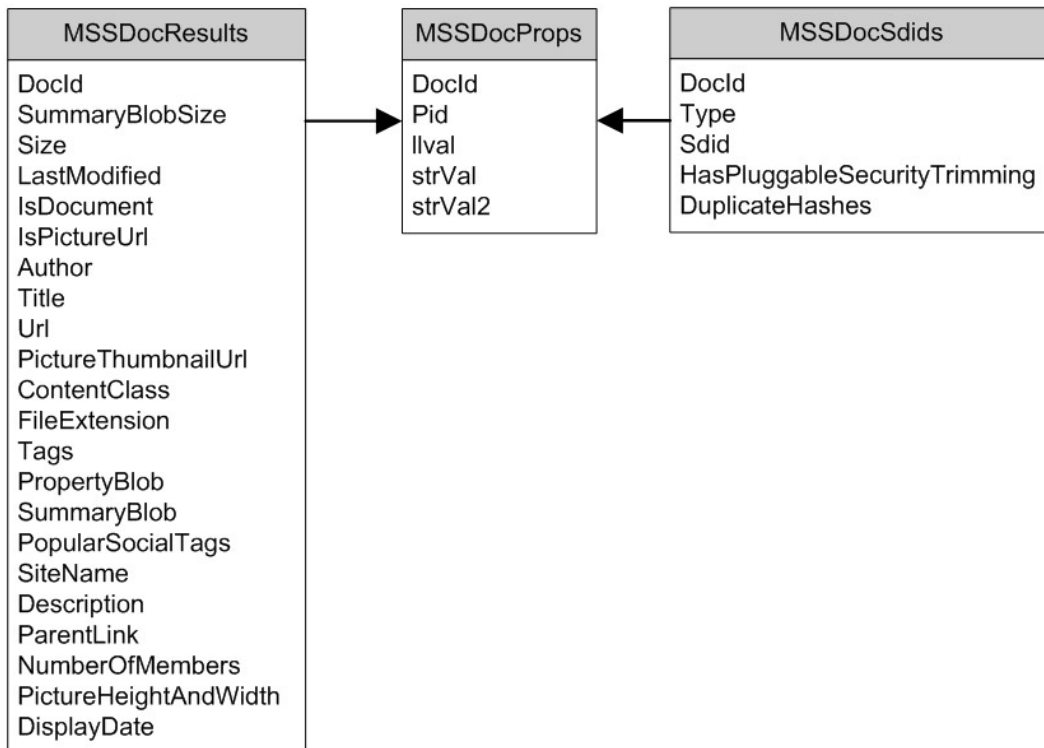


Figure 3: Metadata maintained by a Windows SharePoint Services 4.0 server to support this protocol

The following objects are defined here:

- Best bet set
- Search security descriptor set
- Scopes

3.3.2 Timers

None.

3.3.3 Initialization

Listening endpoints are set up on the protocol server to handle inbound TDS requests. Authentication of the TDS connection to the protocol server MUST occur before this protocol can be used.

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

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

The T-SQL syntax for each Stored Procedure and result set, and the variables they are composed of, is defined in the [\[MSDN-TSQL-Ref\]](#) protocol. In the T-SQL syntax, the variable name is followed by the type of the variable which can optionally have a length value in brackets and can optionally have a default value indicated by an equals sign followed by the default value. Unless otherwise specified, all Stored Procedures defined in this section are located in the Content Database.

For definitional clarity, a name has been assigned to any columns in the Result Sets that do not have a defined name in their current implementation. This does not affect the operation of the Result Set, as the ordinal position of any column with no defined name is expected by the front-end Web server. Such names are designated in the text using curly braces in the form `{name}`.

3.3.5.1 `proc_MSS_FetchSDs`

See section [3.1.5.1](#).

3.3.5.2 `proc_MSS_GetDuplicateHashes`

See section [3.1.5.3](#).

3.3.5.3 `proc_MSS_GetManagedProperties`

See section [3.1.5.5](#).

3.3.5.4 `proc_MSS_GetManagedPropertyAliases`

See section [3.1.5.6](#).

3.3.5.5 proc_MSS_GetMultipleResults

See section [3.1.5.7](#).

3.3.5.6 proc_MSS_GetSchemaHighLevelInfo

See section [3.1.5.11](#).

3.3.5.7 proc_MSS_GetSchemaParameters

See section [3.1.5.12](#).

3.3.5.8 proc_MSS_GetVectorI4ConfigurationProperty

See section [3.1.5.14](#).

3.3.5.9 proc_MSS_PrefetchSDs

See section [3.1.5.15](#).

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

3.4 Windows SharePoint Services 4.0 Client Details

The protocol client role that is served by the search query server role is described in this section. The client role can make requests for query execution and managed property value retrieval operations.

3.4.1 Abstract Data Model

See section [3.2.1](#).

3.4.2 Timers

None.

3.4.3 Initialization

None.

3.4.4 Higher-Layer Triggered Events

None.

3.4.5 Message Processing Events and Sequencing Rules

A user sends a request to the protocol client to perform a search query. In this case, the protocol client performs a list of tasks necessary to return items which match this search query. The tasks

include retrieving items from the protocol server, eliminating certain items based on security information or because of duplicate results, and retrieving managed properties for a list of items using the following sequence:

1. The task of retrieving item properties from the protocol server MUST be accomplished by querying the MSSDocProps (as defined in section [2.2.5.2](#)) or MSSDocSdids (as defined in section [2.2.5.4](#)) tables.
2. The item elimination tasks MUST be accomplished by executing the proc_MSS_FetchSDs (as specified in section [3.1.5.1](#)) and proc_MSS_GetDuplicateHashes (as specified in section [3.3.5.2](#)) stored procedures.
3. The task of retrieving item property values from the protocol server MUST be accomplished by calling proc_MSS_GetMultipleResults (as specified in section [3.3.5.5](#)) to retrieve properties for the query results.

3.4.6 Timer Events

None.

3.4.7 Other Local Events

None.

4 Protocol Examples

This section describes two possible interactions between a client and server through the use of this protocol.

4.1 Query Execution

This example describes the requests made and responses returned when query execution is requested.

Security for this protocol is controlled by the rights to the protocol server (back-end database server), which is negotiated as part of the Tabular Data Stream ([MS-TDS]) protocol.

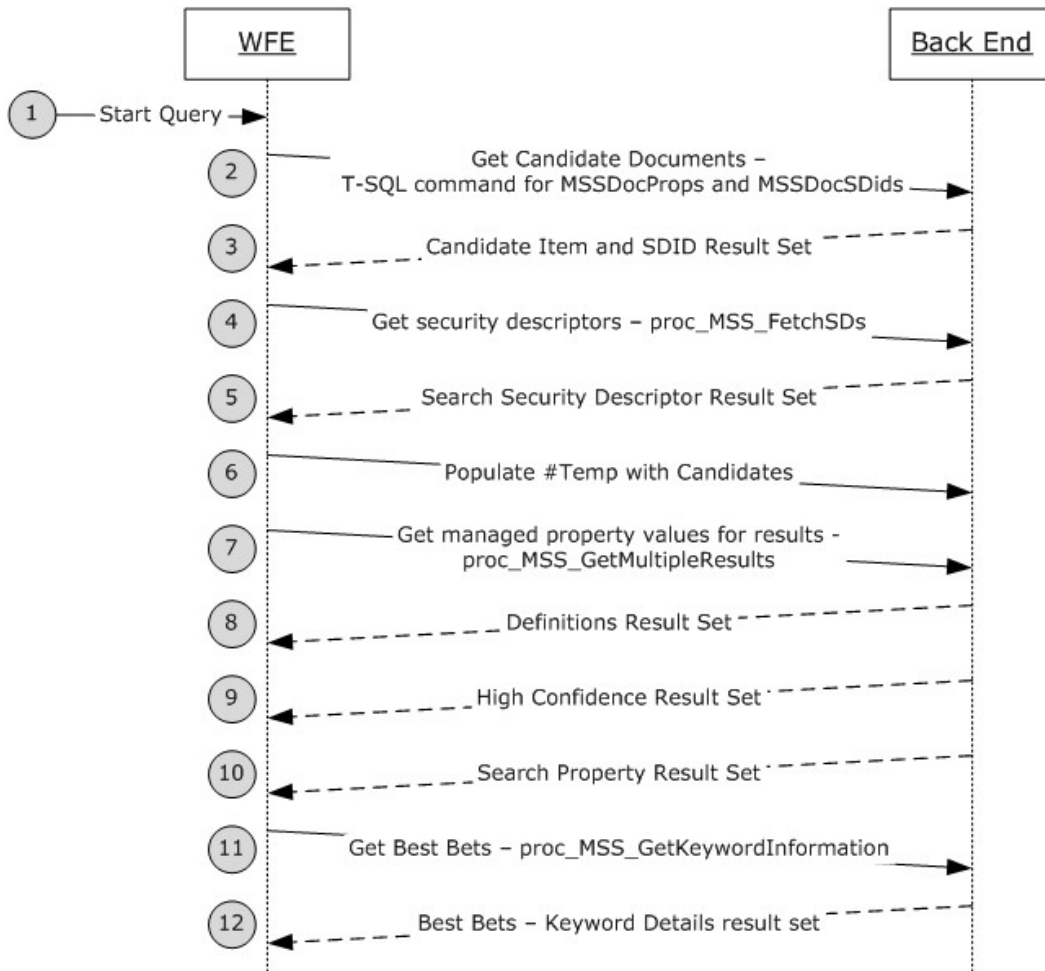


Figure 4: Query execution example

The steps are explained in the following paragraph.

Query execution retrieves candidate documents, prunes the results according to duplication and security information, retrieves properties for the candidate documents and returns the results.

1. The protocol client receives a query request from the user interface or Web Service.

2. The protocol client translates the query into a T-SQL query and sends it to the protocol server (back-end database server).
3. The protocol server returns a set of item identifiers and search security descriptor identifiers which match the T-SQL command.
4. The protocol server returns a Duplication identifier result set which contains duplication identifiers for each item which are the same when those items have similar content and metadata.
5. The protocol client calls **proc_MSS_FetchSDs** to retrieve the search security descriptors that correspond to the items in the query result candidate document set.
6. The protocol server returns a Search Security Descriptors result set for the items specified.
7. The protocol client calls **proc_MSS_GetMultipleResults** to retrieve properties for the query results, specifying 0x80000005 for the *@RequestTypes* parameter. The *@Term* and *@Author* parameters contain strings which the protocol client has determined to be a representative term for the query, and an author, if one was specified in the query. The *@SqlSelect* parameter contains a T-SQL statement which refers to the MSSDocProps table, and which retrieves property values specific to the query.
8. The protocol server returns a Definitions result set for the documents specified in *@Term*.
9. The protocol server returns a High confidence result set for the documents specified in *@Term* and *@Author*.
10. The protocol server returns a Search property result set as per the *@SqlSelect* statement.
11. The protocol client calls **proc_MSS_GetKeywordInformation** to retrieve best bet information, specifying the query term in the *@Term* parameter.
12. The protocol server returns a Keyword details result set for the specified term.

4.2 Query Logging

This example describes the requests made and responses returned during search query logging.

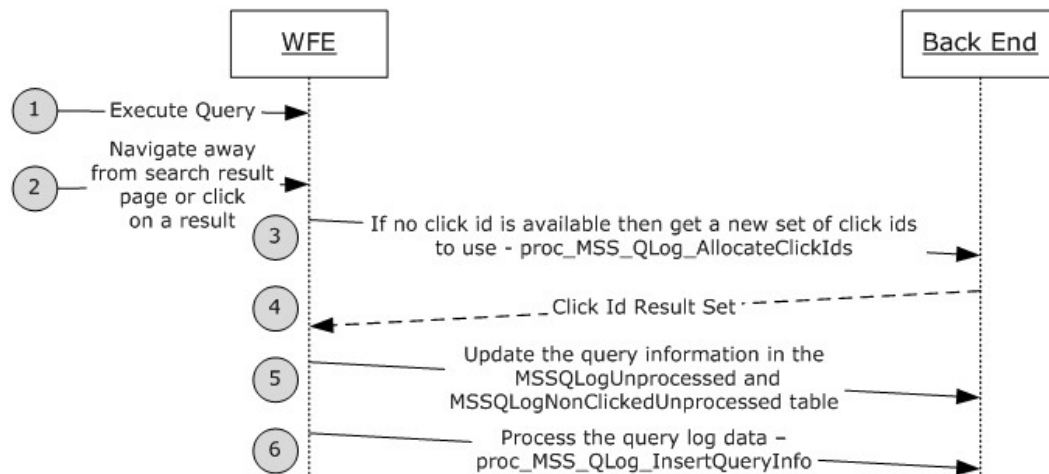


Figure 5: Query logging example

The steps are explained in the following paragraph.

Query Logging logs the information about the query that was executed when the user navigates away from the search results page or when the user clicks on one of the search result.

1. The protocol client completes a query request from the user interface.
2. The protocol client checks to see if it has previously obtained a range of click ids to assign to uniquely identify this query and whether it has not exhausted all the click ids in the range obtained. If it does not have click ids available to be assigned then it makes a request to the protocol server (back-end database server).
3. The protocol server returns a click id result set.
4. The protocol client updates the search query log tables **MSSQLLogUnprocessed** and **MSSQLLogNonClickedUnprocessed**.
5. The protocol client periodically calls the **proc_MSS_QLog_InsertQueryInfo** stored procedure to process the information in the **MSSQLLogUnprocessed** and **MSSQLLogNonClickedUnprocessed** tables.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

- Microsoft® FAST™ Search Server 2010
- Microsoft® SharePoint® Foundation 2010
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

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

- [Best Bet set – server](#) 29
- client ([section 3.2.1](#) 55, [section 3.4.1](#) 59)
- [detailed query log set – server](#) 29
- [historical query log set – server](#) 29
- [query result candidate document set – client](#) 56
- [raw query log set – server](#) 29
- [result document property set – client](#) 56
- [Scopes – server](#) 30
- [search security descriptor set – server](#) 30
- server ([section 3.1.1](#) 28, [section 3.3.1](#) 57)
- [summary query log set – server](#) 30
- [AdvancedSearch flag structure](#) 13
- [Alternative Offset Length Pair binary structure](#) 14
- [Applicability](#) 11
- [Attribute groups - overview](#) 27
- [Attributes - overview](#) 27

B

[Best Bet set](#) 29

Binary structures

- [Alternative Offset Length Pair](#) 14
- [Id Value Pairs Blob](#) 15
- [Id With Value structure](#) 15
- [Offset Length Pair](#) 15
- [Property Values Blob](#) 16
- [Serialized String Values](#) 17
- [Summary Blob](#) 18
- [Binary structures - overview](#) 14
- [Bit fields - overview](#) 13

C

[Capability negotiation](#) 11

[Change tracking](#) 66

Client

- abstract data model ([section 3.2.1](#) 55, [section 3.4.1](#) 59)
- higher-layer triggered events ([section 3.2.4](#) 56, [section 3.4.4](#) 59)
- initialization ([section 3.2.3](#) 56, [section 3.4.3](#) 59)
- local events ([section 3.2.7](#) 57, [section 3.4.7](#) 60)
- message processing ([section 3.2.5](#) 56, [section 3.4.5](#) 59)
- [Microsoft SharePoint Server 2010 interface](#) 55
- [overview - Windows SharePoint Services 4.0 interface](#) 59
- sequencing rules ([section 3.2.5](#) 56, [section 3.4.5](#) 59)
- timer events ([section 3.2.6](#) 57, [section 3.4.6](#) 60)
- timers ([section 3.2.2](#) 56, [section 3.4.2](#) 59)
- [Windows SharePoint Services 4.0 interface](#) 59
- Client - Microsoft SharePoint Server 2010
 - [query result candidate document set](#) 56
 - [result document property set](#) 56

Common data types

- [overview](#) 12
- [Complex types - overview](#) 27
- [Continued flag structure](#) 14

D

Data model - abstract

- [Best Bet set](#) 29
- client ([section 3.2.1](#) 55, [section 3.4.1](#) 59)
- [detailed query log set](#) 29
- [historical query log set](#) 29
- [query result candidate document set](#) 56
- [raw query log set](#) 29
- [result document property set](#) 56
- [Scopes](#) 30
- [search security descriptor set](#) 30
- server ([section 3.1.1](#) 28, [section 3.3.1](#) 57)
- [summary query log set](#) 30

Data types

- [common](#) 12
- [Duplicate Identifier Block simple type](#) 12
- [Id Blob simple type](#) 12
- [LastConsumerChangeID simple type](#) 12
- [LastUpdate simple type](#) 12
- [Managed Property Time simple type](#) 13
- [Managed Type simple type](#) 13
- [Scope simple type](#) 13
- [Session Identifier simple type](#) 13
- [Site Collection Identifier simple type](#) 13
- [Site Identifier simple type](#) 13

Data types - simple

- [Duplicate Identifier Block](#) 12
- [Id Blob](#) 12
- [LastConsumerChangeID](#) 12
- [LastUpdate](#) 12
- [Managed Property Time](#) 13
- [Managed Type](#) 13
- [overview](#) 12
- [Scope](#) 13
- [Session Identifier](#) 13
- [Site Collection Identifier](#) 13
- [Site Identifier](#) 13
- [Detailed query log set](#) 29
- [Duplicate Identifier Block simple type](#) 12

E

[Elements - overview](#) 27

Events

- local - client ([section 3.2.7](#) 57, [section 3.4.7](#) 60)
- local - server ([section 3.1.7](#) 55, [section 3.3.7](#) 59)
- timer - client ([section 3.2.6](#) 57, [section 3.4.6](#) 60)
- timer - server ([section 3.1.6](#) 55, [section 3.3.6](#) 59)

Examples

- [overview](#) 61
- [query execution](#) 61
- [query logging](#) 62

F

[Fields - vendor-extensible](#) 11
Flag structures
 [AdvancedSearch flag](#) 13
 [Continued flag](#) 14
 [ResultView flag](#) 14
[Flag structures - overview](#) 13

G

[Glossary](#) 7
[Groups - overview](#) 27

H

Higher-layer triggered events
 client ([section 3.2.4](#) 56, [section 3.4.4](#) 59)
 server ([section 3.1.4](#) 31, [section 3.3.4](#) 58)
[Historical query log set](#) 29

I

[Id Blob simple type](#) 12
[Id Value Pairs Blob binary structure](#) 15
[Id With Value structure binary structure](#) 15
[Implementer - security considerations](#) 64
[Index of security parameters](#) 64
[Informative references](#) 9
Initialization
 client ([section 3.2.3](#) 56, [section 3.4.3](#) 59)
 server ([section 3.1.3](#) 31, [section 3.3.3](#) 58)
Interfaces - client
 [Microsoft SharePoint Server 2010](#) 55
 [Windows SharePoint Services 4.0](#) 59
Interfaces - server
 [Microsoft SharePoint Server 2010](#) 28
 [Windows SharePoint Services 4.0](#) 57
[Introduction](#) 7

L

[LastConsumerChangeID simple type](#) 12
[LastUpdate simple type](#) 12
Local events
 client ([section 3.2.7](#) 57, [section 3.4.7](#) 60)
 server ([section 3.1.7](#) 55, [section 3.3.7](#) 59)

M

[Managed Property Time simple type](#) 13
[Managed Type simple type](#) 13
Message processing
 client ([section 3.2.5](#) 56, [section 3.4.5](#) 59)
 server ([section 3.1.5](#) 31, [section 3.3.5](#) 58)
Messages
 [AdvancedSearch flag structure](#) 13
 [Alternative Offset Length Pair binary structure](#) 14
 [attribute groups](#) 27
 [attributes](#) 27
 [binary structures](#) 14
 [bit fields](#) 13

[common data types](#) 12
 [complex types](#) 27
 [Continued flag structure](#) 14
 [elements](#) 27
 [enumerations](#) 12
 [flag structures](#) 13
 [groups](#) 27
 [Id Value Pairs Blob binary structure](#) 15
 [Id With Value structure binary structure](#) 15
 [MSSDefinitions structure](#) 20
 [MSSDocProps structure](#) 21
 [MSSDocResults structure](#) 21
 [MSSDocSdids structure](#) 23
 [MSSQLogLocation structure](#) 23
 [MSSQLogLocationSummary structure](#) 24
 [MSSQLogNonClickedUnprocessed structure](#) 24
 [MSSQLogUnprocessed structure](#) 25
 [namespaces](#) 26
 [Offset Length Pair binary structure](#) 15
 [Property Values Blob binary structure](#) 16
 [ResultView flag structure](#) 14
 [Search Security Descriptor result set](#) 20
 [Serialized String Values binary structure](#) 17
 [simple data types](#) 12
 [simple types](#) 27
 [Summary Blob binary structure](#) 18
 [table structures](#) 20
 [transport](#) 12
 [view structures](#) 20
 [XML structures](#) 26
Methods
 proc_MSS_FetchSDs ([section 3.1.5.1](#) 31, [section 3.3.5.1](#) 58)
 proc_MSS_GetDocCountEstimate 32
 proc_MSS_GetDuplicateHashes ([section 3.1.5.3](#) 32, [section 3.3.5.2](#) 58)
 proc_MSS_GetKeywordInformation 33
 proc_MSS_GetManagedProperties ([section 3.1.5.5](#) 34, [section 3.3.5.3](#) 58)
 proc_MSS_GetManagedPropertyAliases ([section 3.1.5.6](#) 41, [section 3.3.5.4](#) 58)
 proc_MSS_GetMultipleResults ([section 3.1.5.7](#) 41, [section 3.3.5.5](#) 59)
 proc_MSS_GetPronunciations 46
 proc_MSS_GetQuerySuggestions 46
 proc_MSS_GetRankingModels 48
 proc_MSS_GetSchemaHighLevelInfo ([section 3.1.5.11](#) 48, [section 3.3.5.6](#) 59)
 proc_MSS_GetSchemaParameters ([section 3.1.5.12](#) 49, [section 3.3.5.7](#) 59)
 proc_MSS_GetStaticRankingFeatures 50
 proc_MSS_GetVectorI4ConfigurationProperty ([section 3.1.5.14](#) 50, [section 3.3.5.8](#) 59)
 proc_MSS_PrefetchSDs ([section 3.1.5.15](#) 51, [section 3.3.5.9](#) 59)
 proc_MSS_QLog_AllocateClickIds 51
 proc_MSS_QLog_InsertQueryInfo 52
 proc_MSS_ScopeCacheGetChangedConsumers 52
 proc_MSS_ScopeCacheGetChanges 53
[Microsoft SharePoint Server 2010 client interface](#) 55

[Microsoft SharePoint Server 2010 server interface](#) 28
[MSSDefinitions structure](#) 20
[MSSDocProps structure](#) 21
[MSSDocResults structure](#) 21
[MSSDocSdids structure](#) 23
[MSSQLogLocation structure](#) 23
[MSSQLogLocationSummary structure](#) 24
[MSSQLogNonClickedUnprocessed structure](#) 24
[MSSQLogUnprocessed structure](#) 25

N

[Namespaces](#) 26
[Normative references](#) 8

O

[Offset Length Pair binary structure](#) 15
Overview
 [query execution](#) 9
 [query logging operations](#) 10
[Overview \(synopsis\)](#) 9

P

[Parameters - security index](#) 64
[Preconditions](#) 11
[Prerequisites](#) 11
[proc_MSS_FetchSDs method \(section 3.1.5.1 31, section 3.3.5.1 58\)](#)
[proc_MSS_GetDocCountEstimate method](#) 32
[proc_MSS_GetDuplicateHashes method \(section 3.1.5.3 32, section 3.3.5.2 58\)](#)
[proc_MSS_GetKeywordInformation method](#) 33
[proc_MSS_GetManagedProperties method \(section 3.1.5.5 34, section 3.3.5.3 58\)](#)
[proc_MSS_GetManagedPropertyAliases method \(section 3.1.5.6 41, section 3.3.5.4 58\)](#)
[proc_MSS_GetMultipleResults method \(section 3.1.5.7 41, section 3.3.5.5 59\)](#)
[proc_MSS_GetPronunciations method](#) 46
[proc_MSS_GetQuerySuggestions method](#) 46
[proc_MSS_GetRankingModels method](#) 48
[proc_MSS_GetSchemaHighLevelInfo method \(section 3.1.5.11 48, section 3.3.5.6 59\)](#)
[proc_MSS_GetSchemaParameters method \(section 3.1.5.12 49, section 3.3.5.7 59\)](#)
[proc_MSS_GetStaticRankingFeatures method](#) 50
[proc_MSS_GetVectorI4ConfigurationProperty method \(section 3.1.5.14 50, section 3.3.5.8 59\)](#)
[proc_MSS_PrefetchSDs method \(section 3.1.5.15 51, section 3.3.5.9 59\)](#)
[proc_MSS_QLog_AllocateClickIds method](#) 51
[proc_MSS_QLog_InsertQueryInfo method](#) 52
[proc_MSS_ScopeCacheGetChangedConsumers method](#) 52
[proc_MSS_ScopeCacheGetChanges method](#) 53
[Product behavior](#) 65
[Property Values Blob binary structure](#) 16
Protocol overview
 [query execution](#) 9

[query logging operations](#) 10

Q

[Query execution example](#) 61
[Query execution overview](#) 9
[Query logging example](#) 62
[Query logging operations overview](#) 10
[Query result candidate document set](#) 56

R

[Raw query log set](#) 29
[References](#) 8
 [informative](#) 9
 [normative](#) 8
[Relationship to other protocols](#) 10
[Result document property set](#) 56
Result sets - messages
 [Search Security Descriptor](#) 20
[ResultView flag structure](#) 14

S

[Scope simple type](#) 13
[Scopes data model](#) 30
[Search Security Descriptor result set](#) 20
[Search security descriptor set](#) 30
Security
 [implementer considerations](#) 64
 [parameter index](#) 64
Sequencing rules
 client ([section 3.2.5 56](#), [section 3.4.5 59](#))
 server ([section 3.1.5 31](#), [section 3.3.5 58](#))
[Serialized String Values binary structure](#) 17
Server
 abstract data model ([section 3.1.1 28](#), [section 3.3.1 57](#))
 higher-layer triggered events ([section 3.1.4 31](#), [section 3.3.4 58](#))
 initialization ([section 3.1.3 31](#), [section 3.3.3 58](#))
 local events ([section 3.1.7 55](#), [section 3.3.7 59](#))
 message processing ([section 3.1.5 31](#), [section 3.3.5 58](#))
 [Microsoft SharePoint Server 2010 interface](#) 28
 [overview - Microsoft SharePoint Server 2010 interface](#) 28
 [overview - Windows SharePoint Services 4.0 interface](#) 57
 [proc_MSS_FetchSDs method \(section 3.1.5.1 31, section 3.3.5.1 58\)](#)
 [proc_MSS_GetDocCountEstimate method](#) 32
 [proc_MSS_GetDuplicateHashes method \(section 3.1.5.3 32, section 3.3.5.2 58\)](#)
 [proc_MSS_GetKeywordInformation method](#) 33
 [proc_MSS_GetManagedProperties method \(section 3.1.5.5 34, section 3.3.5.3 58\)](#)
 [proc_MSS_GetManagedPropertyAliases method \(section 3.1.5.6 41, section 3.3.5.4 58\)](#)
 [proc_MSS_GetMultipleResults method \(section 3.1.5.7 41, section 3.3.5.5 59\)](#)
 [proc_MSS_GetPronunciations method](#) 46

- [proc_MSS_GetQuerySuggestions_method](#) 46
- [proc_MSS_GetRankingModels_method](#) 48
- [proc_MSS_GetSchemaHighLevelInfo_method](#) (section 3.1.5.11 48, section 3.3.5.6 59)
- [proc_MSS_GetSchemaParameters_method](#) (section 3.1.5.12 49, section 3.3.5.7 59)
- [proc_MSS_GetStaticRankingFeatures_method](#) 50
- [proc_MSS_GetVectorI4ConfigurationProperty_method](#) (section 3.1.5.14 50, section 3.3.5.8 59)
- [proc_MSS_PrefetchSDs_method](#) (section 3.1.5.15 51, section 3.3.5.9 59)
- [proc_MSS_OLog_AllocateClickIds_method](#) 51
- [proc_MSS_OLog_InsertQueryInfo_method](#) 52
- [proc_MSS_ScopeCacheGetChangedConsumers_method](#) 52
- [proc_MSS_ScopeCacheGetChanges_method](#) 53
- sequencing rules (section 3.1.5 31, section 3.3.5 58)
- timer events (section 3.1.6 55, section 3.3.6 59)
- timers (section 3.1.2 31, section 3.3.2 58)
- [Windows SharePoint Services 4.0 interface](#) 57

Server - Microsoft SharePoint Server 2010

- [Best Bet set](#) 29
- [detailed query log set](#) 29
- [historical query log set](#) 29
- [raw query log set](#) 29
- [Scopes](#) 30
- [search security descriptor set](#) 30
- [summary query log set](#) 30

[Session Identifier simple type](#) 13

Simple data types

- [Duplicate Identifier Block](#) 12
- [Id Blob](#) 12
- [LastConsumerChangeID](#) 12
- [LastUpdate](#) 12
- [Managed Property Time](#) 13
- [Managed Type](#) 13
- [overview](#) 12
- [Scope](#) 13
- [Session Identifier](#) 13
- [Site Collection Identifier](#) 13
- [Site Identifier](#) 13

[Simple types - overview](#) 27

[Site Collection Identifier simple type](#) 13

[Site Identifier simple type](#) 13

[Standards assignments](#) 11

Structures

- [binary](#) 14
- [table and view](#) 20
- [XML](#) 26

[Summary Blob binary structure](#) 18

[Summary query log set](#) 30

T

Table structures

- [MSSDefinitions](#) 20
- [MSSDocProps](#) 21
- [MSSDocResults](#) 21
- [MSSDocSdids](#) 23
- [MSSQLogLocation](#) 23

- [MSSQLogLocationSummary](#) 24
- [MSSQLogNonClickedUnprocessed](#) 24
- [MSSQLogUnprocessed](#) 25
- [Table structures - overview](#) 20
- Timer events
 - client (section 3.2.6 57, section 3.4.6 60)
 - server (section 3.1.6 55, section 3.3.6 59)
- Timers
 - client (section 3.2.2 56, section 3.4.2 59)
 - server (section 3.1.2 31, section 3.3.2 58)
- [Tracking changes](#) 66
- [Transport](#) 12
- Triggered events - higher-layer
 - client (section 3.2.4 56, section 3.4.4 59)
 - server (section 3.1.4 31, section 3.3.4 58)
- Types
 - [complex](#) 27
 - [simple](#) 27

V

- [Vendor-extensible fields](#) 11
- [Versioning](#) 11
- [View structures - overview](#) 20

W

- [Windows SharePoint Services 4.0 client interface](#) 59
- [Windows SharePoint Services 4.0 server interface](#) 57

X

- [XML structures](#) 26