

[MS-SQLPQ]: Search Service Database Query Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Changes made for template compliance
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	Editorial	Changed language and formatting in the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.6	Minor	Clarified the meaning of the technical content.
04/11/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.6	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	9
1.2.1 Normative References	9
1.2.2 Informative References	9
1.3 Protocol Overview (Synopsis)	9
1.3.1 Property-Based Query Execution & Managed Properties Retrieval Operations	10
1.3.2 Query Logging Operations	11
1.4 Relationship to Other Protocols	12
1.5 Prerequisites/Preconditions	13
1.6 Applicability Statement	13
1.7 Versioning and Capability Negotiation	13
1.8 Vendor-Extensible Fields	13
1.9 Standards Assignments	13
2 Messages	14
2.1 Transport	14
2.2 Common Data Types	14
2.2.1 Simple Data Types and Enumerations	14
2.2.1.1 Site Collection Identifier	14
2.2.1.2 Site Identifier	14
2.2.1.3 Scope	14
2.2.1.4 Advanced Search Flag	14
2.2.1.5 Continued Flag	14
2.2.1.6 DidYouMean Flag	15
2.2.1.7 ResultView Flag	15
2.2.1.8 IsSSPLevel Flag	15
2.2.2 Bit Fields and Flag Structures	15
2.2.3 Binary Structures	15
2.2.4 Common Result Sets	15
2.2.5 Tables and Views	15
2.2.5.1 MSSDocProps	16
2.2.5.2 MSSDocSdids	16
2.2.5.3 MSSDuplicateHashes	17
2.2.5.4 MSSQLogUnprocessed	17
2.2.5.5 MSSQLogNonClickedUnprocessed	19
2.2.5.6 MSSQLogLocation	19
2.2.5.7 #Temp	19
2.2.6 XML Structures	20
3 Protocol Details	21
3.1 Microsoft Office SharePoint Server Details	21
3.1.1 Abstract Data Model	21
3.1.1.1 Managed Property Value Set	22
3.1.1.2 Search Security Descriptor Set	22
3.1.1.3 Item Identifier Set	23
3.1.1.4 Best Bet Set	23
3.1.1.5 Extracted Definition Set	23
3.1.1.6 Scopes	23
3.1.1.7 Raw Query Log Set	24

3.1.1.8	Historical Query Log Set	24
3.1.1.9	Detailed Query Log Set.....	24
3.1.1.10	Summary Query Log Set.....	25
3.1.2	Timers	25
3.1.3	Initialization	25
3.1.4	Message Processing Events and Sequencing Rules.....	26
3.1.4.1	proc_MSS_FetchSDs	28
3.1.4.1.1	Search Security Descriptor Result Set	28
3.1.4.2	proc_MSS_GetConfigurationProperty	28
3.1.4.3	proc_MSS_GetManagedProperties.....	28
3.1.4.4	proc_MSS_GetManagedPropertyAliases	29
3.1.4.4.1	GetManagedPropertyAliases Result Set	29
3.1.4.5	proc_MSS_GetKeywordInformation	29
3.1.4.5.1	Keyword Details Result Set.....	30
3.1.4.6	proc_MSS_GetMultipleResults	30
3.1.4.6.1	Definitions Result Set	31
3.1.4.6.2	High Confidence Result Set.....	31
3.1.4.6.3	Search Property Result Set.....	32
3.1.4.7	proc_MSS_GetSchemaHighLevelInfo	33
3.1.4.8	proc_MSS_GetSchemaParameters	33
3.1.4.9	proc_MSS_GetVectorI4ConfigurationProperty.....	33
3.1.4.9.1	Vector Configuration Property Result Set.....	33
3.1.4.10	proc_MSS_QLog_AllocateClickIds.....	34
3.1.4.10.1	Click Id Result Set	34
3.1.4.11	proc_MSS_QLog_DeleteOld	34
3.1.4.12	proc_MSS_QLog_InsertQueryInfo	35
3.1.4.13	proc_MSS_QLog_LowClickThrough	35
3.1.4.13.1	Low ClickThrough Result Set	36
3.1.4.14	proc_MSS_QLog_QueriesByDay	36
3.1.4.14.1	Queries By Day Result Set	37
3.1.4.15	proc_MSS_QLog_QueriesByMonth.....	37
3.1.4.15.1	Queries By Month Result Set.....	38
3.1.4.16	proc_MSS_QLog_QueriesByScope	38
3.1.4.16.1	Queries By Scope Result Set	39
3.1.4.17	proc_MSS_QLog_TopBestBets	39
3.1.4.17.1	Top Best Bets Result Set	39
3.1.4.18	proc_MSS_QLog_TopDestinations	40
3.1.4.18.1	Top Destinations Result Set	40
3.1.4.19	proc_MSS_QLog_TopQueries.....	41
3.1.4.19.1	Top Queries Result Set.....	41
3.1.4.20	proc_MSS_QLog_TopSiteCollections	42
3.1.4.20.1	Top Site Collections Result Set	42
3.1.4.21	proc_MSS_QLog_ZeroBestBets	43
3.1.4.21.1	Zero best bets Result Set	43
3.1.4.22	proc_MSS_QLog_ZeroResults	43
3.1.4.22.1	Zero Results Result Set	44
3.1.4.23	proc_MSS_RetrieveDuplicateHashes.....	45
3.1.4.23.1	Duplicate Hashes Result Set	45
3.1.4.24	proc_MSS_RetrieveDuplicateHashesEx	45
3.1.4.24.1	Duplicate HashesEx Result Set	46
3.1.4.25	proc_MSS_ScopeCacheGetChangedConsumers	46
3.1.4.25.1	Changed Consumers Result Set.....	46
3.1.4.25.2	Last Consumer Change ID Result Set.....	46

3.1.4.26	proc_MSS_ScopeCacheGetChanges	47
3.1.4.26.1	Scopes Result Set.....	47
3.1.4.26.2	Display Groups Result Set	48
3.1.4.26.3	Compiled Scopes Result Set.....	48
3.1.4.26.4	Last Update Result Set.....	48
3.1.5	Timer Events	48
3.1.6	Other Local Events	49
3.2	Client Details for Windows Sharepoint Server	49
3.2.1	Abstract Data Model	49
3.2.1.1	Query Result Candidate Document Set	49
3.2.1.2	Result Document Property Set.....	49
3.2.2	Timers	50
3.2.3	Initialization	50
3.2.4	Message Processing Events and Sequencing Rules.....	50
3.2.5	Timer Events	50
3.2.6	Other Local Events	50
3.3	Windows SharePoint Services Server Details	50
3.3.1	Abstract Data Model	50
3.3.1.1	Managed property value set.....	51
3.3.1.2	Search security descriptor set	51
3.3.1.3	Item identifier set.....	51
3.3.2	Timers	51
3.3.3	Initialization	51
3.3.4	Message Processing Events and Sequencing Rules.....	52
3.3.4.1	proc_MSS_FetchSDs	53
3.3.4.2	proc_MSS_GetConfigurationProperty.....	53
3.3.4.3	proc_MSS_GetManagedProperties.....	53
3.3.4.4	proc_MSS_GetManagedPropertyAliases	53
3.3.4.5	proc_MSS_GetMultipleResults	53
3.3.4.5.1	Search Property Result Set.....	54
3.3.4.6	proc_MSS_GetSchemaHighLevelInfo	54
3.3.4.7	proc_MSS_GetSchemaParameters	54
3.3.4.8	proc_MSS_GetVectorI4ConfigurationProperty.....	55
3.3.4.8.1	Vector Configuration Property Result Set.....	55
3.3.4.9	proc_MSS_RetrieveDuplicateHashes.....	55
3.3.4.10	proc_MSS_RetrieveDuplicateHashesEx	55
3.3.5	Timer Events	55
3.3.6	Other Local Events	55
3.4	Protocol Client (front-end web server) details for Windows SharePoint Services Server.....	55
3.4.1	Abstract Data Model	56
3.4.2	Timers	56
3.4.3	Initialization	56
3.4.4	Message Processing Events and Sequencing Rules.....	56
3.4.5	Timer Events	56
3.4.6	Other Local Events	56
4	Protocol Examples.....	57
4.1	Query Execution	57
4.2	Query Logging and Usage Reporting.....	59
5	Security.....	62
5.1	Security Considerations for Implementers.....	62

5.2 Index of Security Parameters	62
6 Appendix A: Product Behavior	63
7 Change Tracking.....	64
8 Index	65

1 Introduction

This document specifies the communication sequences that are used by the protocol client (Web and application servers) to perform data query and update commands on the protocol server (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 [\[MS-GLOS\]](#):

Coordinated Universal Time (UTC)
GUID
HRESULT
Hypertext Transfer Protocol (HTTP)
Security Support Provider Interface (SSPI)
Windows security descriptor

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

back-end database server
best bet
Boolean
configuration property
content database
contextual search scope
datetime
default search scope
document identifier
duplicate result removal
extracted definition
federated location
front-end Web server
high confidence property
high confidence results
item
keyword
keyword synonym
managed property
metadata index
metadata schema
pluggable security authentication
property identifier
query result
query server
query text
rank
restriction
result set
return code
search application

search database
search query
search query log
search scope
search scope compilation
search scope consumer
search scope display group
search scope rule
search security descriptor
search shared application object
Shared Services Provider (SSP)
site
site collection
sort order
stored procedure
Transact-Structured Query Language (T-SQL)
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
variant type
vector configuration property

The following terms are specific to this document:

click identifier: A 64-bit integer that uniquely identifies an entry in a query log, corresponding to a single instance of a specific search result being clicked or not clicked for a specific search query.

click ratio: The percentage of times that any search result was clicked for a given term in query text.

clickthrough rate: The total number of query results that are clicked for a search query, divided by the total number of query results that are clicked for all search queries that occurred during the past thirty days.

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

extracted term: A term that an extracted definition applies to.

logged search query: The query text, search scopes, and contextual scope in which a query was executed.

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.

result position: An integer that identifies the relative location of a relevant result within a search results page. The first relevant result has a result position of 1, the second has 2, the 11th has 11, and so forth.

search query log report: A report that is generated from query log information. For example, a search query log report might include the number of queries that were executed every day over the past thirty days.

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

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

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

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

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

1.2.2 Informative References

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

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

1.3 Protocol Overview (Synopsis)

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

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 front-end Web server.

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

- Microsoft Office Sharepoint Server 2007 search query role: This role serves the search query requests for an Microsoft Office Sharepoint Server 2007 **search application**. This role serves requests for search query execution, managed properties retrieval operations and search query log operations.

- Windows SharePoint Services search query role: This role serves the search query request for a Windows SharePoint Services search application. This role only serves requests for search query execution and managed properties retrieval operations.

1.3.1 Property-Based Query Execution & Managed Properties Retrieval Operations

This protocol enables a search application to provide methods for retrieving **items** 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 following diagram shows the data flow between the protocol client (front-end Web server) and the protocol server (back-end database server) with regards to search query execution and managed properties retrieval.

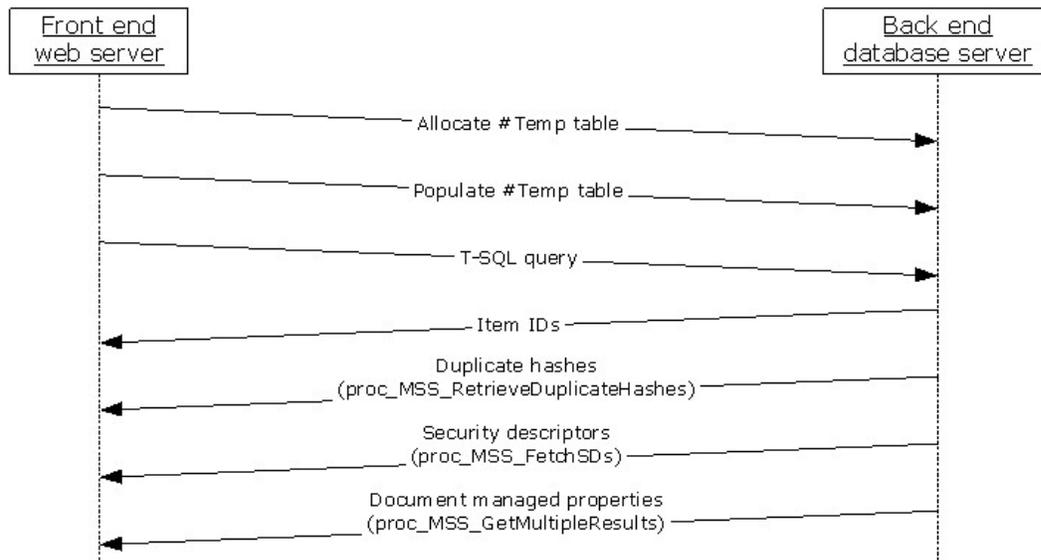


Figure 1: Query Execution Operation Data Flow Model

In the preceding figure, the data flow for the protocol, the protocol client initiates search query execution by allocating the **#Temp** (section 2.2.5.7) table and optionally populates it with **document identifiers** that have been determined to be a subset of the full **relevant results**. Finally, the client issues a **Transact-Structured Query Language (T-SQL)** query to the server. The T-SQL query can be used to retrieve

- Document identifiers for a list of items that match the specified T-SQL query
- **Best bets** that match the terms in the specified T-SQL query
- **High confidence results** that match the terms in the specified T-SQL query

This protocol also allows clients to retrieve managed properties for a list of items. The protocol client starts by allocating a temporary table at the protocol server. The client then makes a call to fill the temporary table with document identifiers. The client can then retrieve an identifier of the items, the **search security descriptors** of the items, and managed properties of the items.

1.3.2 Query Logging Operations

This protocol enables a Microsoft Office SharePoint Server 2007 search application to provide methods for adding, modifying, and deleting information from the search query log.

The following figure shows the data flow between the protocol client (front-end Web server) and the protocol server (back-end database server) during search query log operations.

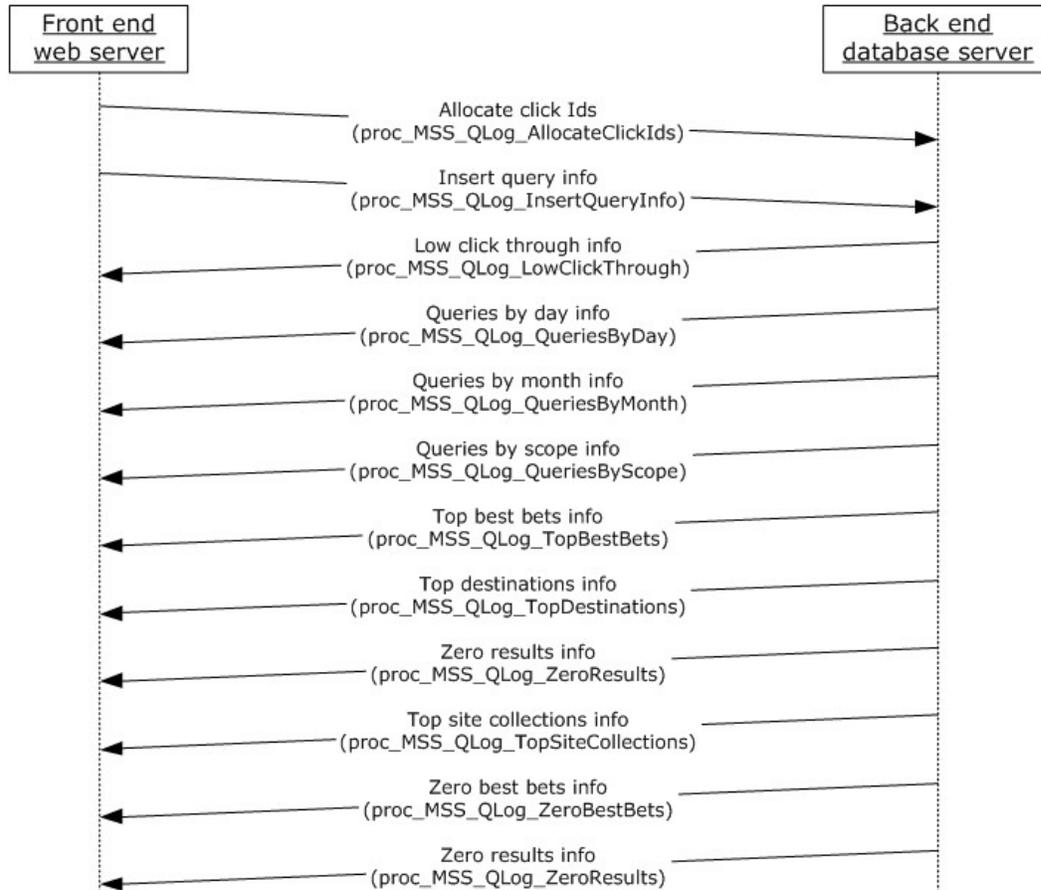


Figure 2: Query Logging Operation Data Flow Diagram

In the preceding figure, 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 the actions of the user on the search results page.

- If the **user** clicks on at least one **query result**, then the following list of information is added to the search query log.
- Information about search query
- 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.

The list of information is added to the search query log every time the user clicks on a search result.

- If the user navigates away from the search results page without clicking on any search results, all the information about the search query is added to the search query log.

The protocol also allows the client to query the search query log for **search query log reports**. The client can request:

- List of search queries with a low **clickthrough rate**
- List of the number of search queries executed every day over the past 30 days
- List of the number of search queries executed every month over the past 12 months
- List of the number of search queries executed in every **search scope** over the past 30 days
- List of the most clicked best bets over the past 30 days
- List of the most common search results pages over the past 30 days
- List of the most common **site collections** from which search queries were executed over the past 30 days
- List of the most common **query text** with zero best bets over the past 30 days
- List of the most common search queries with zero search results over the past 30 days

1.4 Relationship to Other Protocols

This protocol relies on the Tabular Data Stream Protocol (see [\[MS-TDS\]](#)) as its transport protocol to call **stored procedures** to inspect and manipulate item properties by means of **result sets** and **return codes**.

This relationship is illustrated in the following diagram.

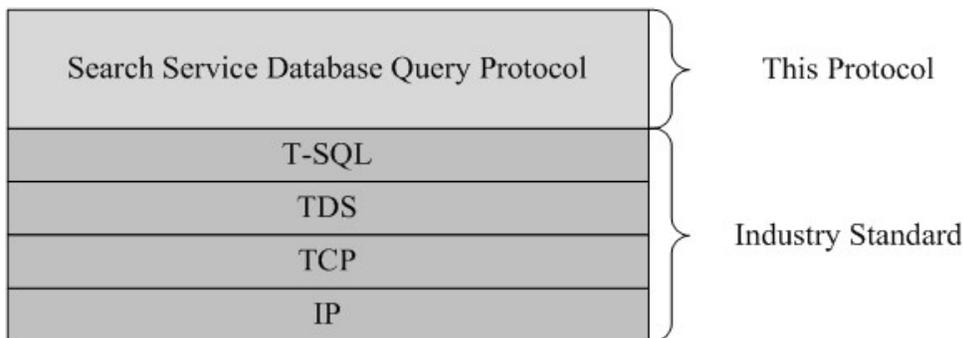


Figure 3: This protocol in relation to other protocols

This protocol relies on the Tabular Data Stream Protocol (see [\[MS-TDS\]](#)) as its transport protocol to call stored procedures to inspect and manipulate **metadata schema** by means of 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 tables are present in the metadata index that is being queried on the back-end database server. The tables in the metadata index contain valid data in a consistent state to be queried successfully by the stored procedures.

1.6 Applicability Statement

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

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.

This document covers versioning issues in the following areas:

Security and Authentication Methods

This protocol supports **SSPI** and SQL Authentication with the back-end database server. These authentication methods are described 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 the 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 simple data types are defined in this specification.

2.2.1.1 Site Collection Identifier

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

2.2.1.2 Site Identifier

A **site** identifier is a **GUID** that uniquely identifies a site within a site collection.

2.2.1.3 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.

2.2.1.4 Advanced Search Flag

The **Advanced Search** flag identifies usage of the Advanced search page. The Advanced search page is a web page that enables end users numerous options to specify additional criteria for their search query. It MUST be a 1-bit number of type bit. The **Advanced Search** flag bit MUST have one of the values listed in the following table.

Value	Meaning
0	The search query was executed from the Advanced search page.
1	The search query was not executed from the Advanced search page.

2.2.1.5 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. The **Continued** flag bit MUST have one of the values listed in the following table.

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

2.2.1.6 DidYouMean Flag

The **DidYouMean** flag identifies whether the spelling suggestion was returned for a search query. It MUST be a 1-bit number of type bit. The **DidYouMean** flag bit MUST have one of the values listed in the following table.

Value	Meaning
0	The spelling suggestion was not returned for this search query.
1	The spelling suggestion was returned for this search query.

2.2.1.7 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 of relevant results is in descending order of their rank .
1	The sort order of relevant results is in descending order of their managed property modified date.

2.2.1.8 IsSSPLevel Flag

The **IsSSPLevel** flag specifies the scope of a search query log report. It MUST be a 32-bit number of type int. The **IsSSPLevel** flag MUST have one of the values listed in the following table.

Value	Meaning
0	The information in the search query log report is scoped to a specific site (or site collection).
1	The search query log report contains information for the entire Shared Services Provider (SSP) .

2.2.2 Bit Fields and Flag Structures

None.

2.2.3 Binary Structures

None.

2.2.4 Common Result Sets

None.

2.2.5 Tables and Views

The following tables are defined in this specification.

2.2.5.1 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,
    RowId          smallint NOT NULL,
    llVal         bigint NULL,
    strVal        nvarchar(64) NULL,
    binVal        image NULL,
    fltVal        float NULL
);
```

DocId: The unique identifier of an item.

Pid: The unique identifier of a managed property.

RowId: The unique identifier for rows with the same **DocId** and **Pid**.

llVal: The numeric value of the managed property. It MUST not be NULL if the **variant type** of the value of the managed property is not **integer**, **Boolean**, or **string**. It holds a hash of the strVal column, if **strVal** is not NULL. It MUST be NULL otherwise.

strVal: The string value of the managed property. If the value length is greater than the size of **strVal**, then the overflow is stored in **binVal**. It MUST be NULL if the variant type of the value of the managed property is not of type **string**.

binVal: The overflow from **strVal**. It MUST NOT be NULL if value length of the string property is greater than length of the strVal column. MUST be NULL otherwise.

fltVal: The floating-point numeric value of the managed property. It MUST be NULL if the variant type of the value of the managed property is not of type **floating-point**.

2.2.5.2 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
);
```

DocId: The identifier of an item.

Type: The identifier of the type of the search security descriptor. MUST be 0 when the search security descriptor of the item is a **Windows security descriptor**. Otherwise, it MUST be 1.

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

HasPluggableSecurityTrimming: Indicates whether item uses **pluggable security authentication**. If true, MUST be 1. Otherwise, it MUST be 0.

2.2.5.3 MSSDuplicateHashes

The **MSSDuplicateHashes** table stores the identifier of item's that are used for **duplicate result removal**. For any given item there MUST be exactly 6 rows in the **MSSDuplicateHashes** table. Each of these 6 rows SHOULD contain an identifier of the item.

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

```
TABLE MSSDuplicateHashes (
    DocId          int NOT NULL,
    HashVal        bigint NOT NULL
);
```

DocId: The unique identifier of an item.

HashVal: The identifier of the item.

2.2.5.4 MSSQLLogUnprocessed

The **MSSQLLogUnprocessed** table stores information about each search query and clicked query results until the information is processed by the stored procedure **proc_MSS_QLog_InsertQueryInfo** (section [3.1.4.12](#)). Each row in this table corresponds to either a search query only or clicked query result for a search query. The stored procedure **proc_MSS_QLog_InsertQueryInfo** (section [3.1.4.12](#)) truncates this table after processing all the information in it.

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,
    searchData      smalldatetime NULL,
    clickedUrlRank   int NULL,
    numResults       int NULL,
    numHighConf      int NULL,
    numBestBets      int NULL,
    numScopes        int NULL,
    resultView       tinyint NULL,
    advancedSearch   bit NULL,
    didYouMean       bit NULL,
    continued        bit NULL,
    contextualScope  nvarchar(100) NULL,
    contextualScopeUrl nvarchar(1024) NULL,
    location         nvarchar(60) NULL
```

);

clickId: This is used to uniquely identify an entry in search the query log. See **Query Logging Operations** as specified in section [1.3.2](#).

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 the 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 [2.2.1.3](#).

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

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

siteGUID: The GUID of the site (or of the site collection) from which the search query was executed. This MUST be a **Site Identifier** as specified in section [2.2.1.2](#) or a **Site Collection Identifier** as specified in section [2.2.1.1](#).

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

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

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

clickedUrlRank: The result position of **clickedUrl**. This MUST be NULL if **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 **ResultView flag**, as specified in section [2.2.1.7](#).

advancedSearch: The value MUST be an **Advanced Search Flag**, as specified in section [2.2.1.4](#).

didYouMean: The value MUST be a **DidYouMean Flag**, as specified in section [2.2.1.6](#).

continued: The value MUST be a **Continued Flag**, as specified in section [2.2.1.5](#).

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

contextualScopeUrl: The URI of the contextualScope.

location: [<1>](#) The name of the **federated location** where the search query was executed.

2.2.5.5 MSSQLLogNonClickedUnprocessed

The **MSSQLLogNonClickedUnprocessed** table stores information about a non-clicked relevant result until it is processed by the stored procedure **proc_MSS_QLog_InsertQueryInfo** (section [3.1.4.12](#)). Each row in this table corresponds to a non-clicked URI. The stored procedure **proc_MSS_QLog_InsertQueryInfo** (section [3.1.4.12](#)) truncates this table after processing the information in it.

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. See **Query Logging Operations** as specified in section [1.3.2](#).

rank: The result position of the non-clicked URI.

nonClickedUrl: The non-clicked relevant result's URI. See **Query Logging Operations** as specified in section [1.3.2](#).

2.2.5.6 MSSQLLogLocation

The **MSSQLLogLocation** table stores the time at which a search query was executed in a federated location.

```
TABLE MSSQLLogLocation(  
    locationId       int NOT NULL,  
    location         nvarchar(60) NULL,  
    lastReference    datetime NULL  
);
```

locationId: The unique identifier of the federated location.

location: The unique internal name of the federated location.

lastReference: The **Coordinated Universal Time (UTC)** datetime when a search query was executed in the federated location.

2.2.5.7 #Temp

The **#Temp** temporary table keeps document identifiers of a list of items. The stored procedures that use the data in this table **MUST** delete all the rows before returning. The **#Temp** table is automatically created by the higher layer of each back-end database server session and **MUST NOT** be created by any other means.

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

```
TABLE #Temp(  
    docId            int NOT NULL,  
    value            int NULL
```

);

docId: The unique identifier of an item.

value: A value provided by the protocol client. The semantics of the value depend on the T-SQL statement that makes use of the temporary table **#Temp**.

2.2.6 XML Structures

None.

3 Protocol Details

3.1 Microsoft Office SharePoint Server Details

The Microsoft Office SharePoint server role is described in this section. The Microsoft Office SharePoint Server 2007 role serves requests for search query execution, managed properties retrieval operations, and search query log 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 back-end database server maintains the following sets of data for this protocol within the metadata schema. Data within the back-end database server are maintained until updated or removed. The following objects are defined here:

- Managed property value set
- Search security descriptor set
- Item Identifier set used for duplicate result removal
- Extracted definition set
- Scopes
- Raw Query Log set
- Historical Query Log set
- Detailed Query Log set
- Summary Query Log set

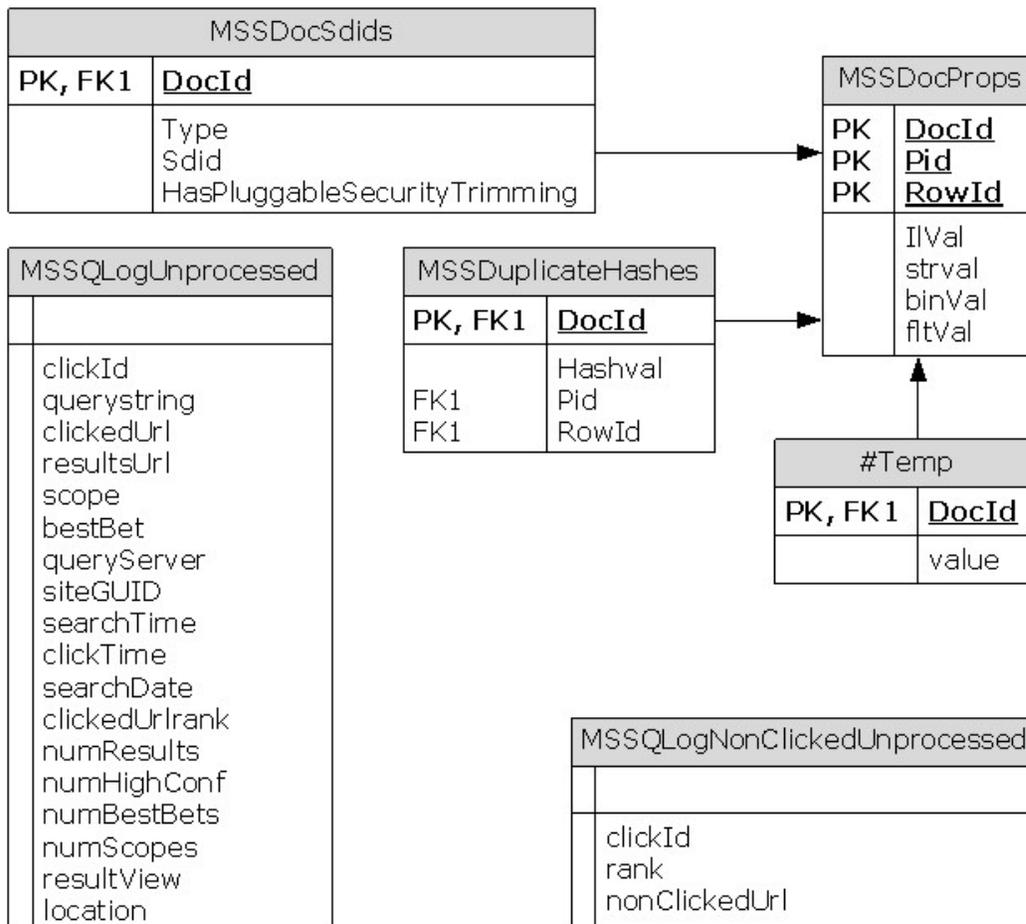


Figure 4: Query Execution and Query Logging Abstract Model

The following subsections define these data structures.

3.1.1.1 Managed Property Value Set

The managed property value set maintains the managed property values for each item. Each managed property value has the following attributes:

- **Property identifier**
- Document identifier
- Managed property value

3.1.1.2 Search Security Descriptor Set

The search security descriptor set maintains security information for each item. The search security descriptor set **MUST** have the following attributes:

- Document identifier
- The search security descriptor type

- The search security descriptor value.

3.1.1.3 Item Identifier Set

The item identifier set contains six identifiers for each item. The identifiers are calculated from item contents and metadata and are used to perform duplicate result removal. The item identifier set has the following attributes:

- Document identifier
- Identifiers of item used for duplicate result removal
- Identifier1
- Identifier2
- Identifier3
- Identifier4
- Identifier5
- Identifier6

3.1.1.4 Best Bet Set

The best bet set contains definitions, **keyword synonyms**, and suggested links for best bets configured by administrators. The best bet set has the following attributes:

- Keyword for which the best bet is defined
- Definition of the keyword
- Any number of suggested links, including
- Title of the best bet
- URI of the best bet.
- Description

3.1.1.5 Extracted Definition Set

The extracted definition set contains **extracted terms** and the corresponding **extracted definitions**. The extracted definition set also contains the document identifier of the item where the definition was found. The extracted definition set has the following attributes:

- Extracted term
- Document identifier of the item which contains the definition
- Extracted definition
- The position of the extracted term in the extracted definition

3.1.1.6 Scopes

For details see [\[MS-SQLPADM\]](#).

3.1.1.7 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 (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:** [<2>](#) The name of the federated location where the search query was executed.

3.1.1.8 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 (or of the site collection) from which the search query was executed.

3.1.1.9 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 (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:** [<3>](#) The name of the federated location where the **search query** was executed.

3.1.1.10 Summary Query Log Set

The summary query log set maintains a datewise 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 (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.
- Number of times the search query was executed.
- **Results Page URL:** The URI of the search results page.
- Number of times zero results were returned for the search query.
- Number of times zero best bets were returned for the query text
- URIs clicked in the results page.
- **Best bets:** The best bets that were clicked in the results page.
- **Federated Location:** [<4>](#) The name of the federated location where the search query was executed.

3.1.2 Timers

An execution timeout timer on the protocol server governs the execution time for the client's requests. The amount of time is specified by a timeout value that is configured on the protocol server for all connections.

3.1.3 Initialization

Listening endpoints are set up on the back-end database server to handle inbound TDS requests.

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

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

3.1.4 Message Processing Events and Sequencing Rules

Unless otherwise specified, all stored procedures defined in this section are located in the **search 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}`.

Unless otherwise specified, all stored procedure input parameters MUST NOT be NULL. As stored procedures use the input parameters for data retrieval from tables, failure to provide valid values will (unless otherwise specified) cause an error as specified in [\[MS-TDS\]](#) that MUST be handled appropriately by the protocol client or the system behavior is indeterminate.

The following table summarizes the stored procedures available in Microsoft Office SharePoint Server 2007 search application for search query execution and managed properties retrieval operations, that are defined in this specification:

Procedure Name	Description
proc_MSS_FetchSDs	Retrieves search security descriptors for the specified list of items.
proc_MSS_GetConfigurationProperty (specified in [MS-SQLPGAT])	Gets the value of a property of the search shared application object or the configuration property structure.
proc_MSS_GetManagedProperties (specified in [MS-SQLPGAT])	Lists managed properties from the metadata schema which were added or modified on or after the given time.
proc_MSS_GetManagedPropertyAliases (specified in [MS-SQLPGAT])	Lists the aliases for managed properties from the metadata schema which were added or modified on or after the given time.
proc_MSS_GetKeywordInformation	Retrieves the definition and best bet for the specified keyword .
proc_MSS_GetMultipleResults	Retrieves extracted definitions for the specified term. Also, retrieves high confidence results that match the specified query. Also, retrieves values for specified managed properties for the specified list of items.
proc_MSS_GetSchemaHighLevelInfo (specified in [MS-SQLPGAT])	Retrieves last modified and last deleted timestamps from the metadata schema.
proc_MSS_GetSchemaParameters (specified in [MS-SQLPGAT])	Retrieves a list of schema parameters from the metadata schema.
proc_MSS_GetVectorI4ConfigurationProperty	Gets the value of the vector configuration property structure from the search shared application object or the configuration property structure.
proc_MSS_RetrieveDuplicateHashes	Retrieves identifiers for the list of items specified in the temporary table #Temp

Procedure Name	Description
	(section 2.2.5.7). The identifiers are used for duplicate results removal.
proc_MSS_RetrieveDuplicateHashesEx	Retrieves identifiers for the specified list of items. The identifiers are used for duplicate results removal.
proc_MSS_ScopeCacheGetChangedConsumers	Retrieves 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.
proc_MSS_ScopeCacheGetChanges	Retrieves search scopes, search scope display groups, and membership information for the specified search scope consumer.

The following table summarizes the stored procedures available in Microsoft Office SharePoint Server 2007 search application for search query log operations, that are defined in this specification:

Procedure Name	Description
proc_MSS_QLog_AllocateClickIds	Identifies a list of click identifiers that can be used to uniquely log search query information.
proc_MSS_QLog_DeleteOld	Clears out data from the search query log that was logged before a specific time period.
proc_MSS_QLog_InsertQueryInfo	Processes the raw query log set.
proc_MSS_QLog_LowClickThrough	Retrieves the top 300 logged search queries with the lowest clickthrough rate in the past 30 days.
proc_MSS_QLog_QueriesByDay	Lists the number of search queries issued per day for the past 30 days.
proc_MSS_QLog_QueriesByMonth	Lists the number of search queries issued per month for the past 12 months.
proc_MSS_QLog_QueriesByScope	Lists the number of search queries issued under each search scope in the past 30 days.
proc_MSS_QLog_TopBestBets	Retrieves the 300 best bets that had the most number of clicks over the past 30 days.
proc_MSS_QLog_TopDestinations	Retrieves URIs of relevant results with the most number of clicks over the past 30 days.
proc_MSS_QLog_TopQueries	Retrieves the logged search queries issued the most number of times in the past 30 days.
proc_MSS_QLog_TopSiteCollections	Retrieves the total number of search queries executed on each site collection in the current SSP over the past 30 days.
proc_MSS_QLog_ZeroBestBets	Retrieves the top 300 query texts with zero best bets in the past 30 days.
proc_MSS_QLog_ZeroResults	Retrieves the top 300 logged search queries with zero relevant

Procedure Name	Description
	results in the past 30 days.

3.1.4.1 proc_MSS_FetchSDs

The **proc_MSS_FetchSDs** stored procedure is called to retrieve the search security descriptors for a list of items. The identifiers of the search security descriptors to be returned MUST be stored in the **value** field of the temporary table **#Temp** (section [2.2.5.7](#)).

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

```
PROCEDURE proc_MSS_FetchSDs (
    @type          int NOT NULL
);
```

@ type: The type of the search security descriptor that MUST be included in the result set. This MUST be 0 when the search security descriptor of the item is a Windows security descriptor. Otherwise, it MUST be 1.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.1.1 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 document identifier and the corresponding search security descriptor of an item. The result set MUST contain zero or more rows. The result set MUST NOT have more rows than the number of rows present in the temporary table **#Temp** (section [2.2.5.7](#)). All the rows in the predefined temporary table **#Temp** (section [2.2.5.7](#)) MUST be removed after retrieving the search security descriptors.

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

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

sdid: The identifier of the search security descriptor.

sd: A binary image that contains the search security descriptor of the item. The mechanism for populating this field is described in [\[MS-SQLPGAT\]](#). The semantics of the data stored in this field are specific to the protocol client.

3.1.4.2 proc_MSS_GetConfigurationProperty

For details see [\[MS-SQLPGAT\]](#).

3.1.4.3 proc_MSS_GetManagedProperties

For details see [\[MS-SQLPGAT\]](#).

3.1.4.4 proc_MSS_GetManagedPropertyAliases

The **proc_MSS_GetManagedPropertyAliases** stored procedure is called to list the aliases for managed properties from the metadata schema that 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: The earliest per-item "last modified" 8-byte datetime for including a managed property in the result set.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.4.1 GetManagedPropertyAliases Result Set

The **GetManagedPropertyAliases** 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 for a managed property.

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

LastModified: This parameter MUST be ignored by the client.

3.1.4.5 proc_MSS_GetKeywordInformation

The **proc_MSS_GetKeywordInformation** stored procedure is called to retrieve the definition and best bets defined for a keyword.

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 keyword, for which the definition and or best bets are requested.

@ConsumerGpId: The identifier of a consumer group from which the best bet details MUST be retrieved.

@Keyword: The keyword whose definition and best bets, if available, MUST be in the result set. If no definition is available then this MUST be set to NULL.

@Definition: The definition of the keyword, if available.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.5.1 Keyword Details Result Set

The Keyword Details result set contains information about a list of best bets defined by the administrator for the specified *@Term*. Each row in the result set contains information about a best bet. The rows in the result set MUST be in the ascending order of relative position of the best bet among all best bets for a specified keyword. The result set contains zero rows if no best bets are available for the specified *@Term*.

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

```
Title          nvarchar(100),
Url            nvarchar(2048),
Description    nvarchar(500);
```

Title: Title of the best bet.

Url: URI of the best bet.

Description: Description of the best bet.

3.1.4.6 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 list of managed properties are specified in *@SqlSelect*. The list of items is stored in the temporary table **#Temp** (section [2.2.5.7](#)). If the temporary table is not available then the list of documents MUST be specified in *@SqlSelect*.

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

```
PROCEDURE proc_MSS_GetMultipleResults (
    @RequestTypes    int,
    @Term            NVARCHAR(100),
    @Author          NVARCHAR(100),
    @SqlSelect       NVARCHAR(2048)
);
```

@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 be set to one of the following values.

Value	Description
0x00000001	Managed properties for a list of items MUST be returned.
0x00000004	The high confidence results that match <i>@Term</i> MUST be returned
0x80000000	The extracted definition for <i>@Term</i> MUST be returned

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

@Author: All the **restrictions** imposed on the author managed property.

@SqlSelect: The T-SQL statement that MUST be executed to get the managed properties for a list of items. The document identifiers of the items MUST be stored in the **docId** field of the temporary table **#Temp** (section [2.2.5.7](#)) or specified in the *@SqlSelect* statement. The T-SQL statement MUST retrieve data from the **MSSDocProps** (section [2.2.5.1](#)), **MSSDocSdids** (section [2.2.5.2](#)) and **#Temp** (section [2.2.5.7](#)) tables.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return zero or one of each of the following result sets:

3.1.4.6.1 Definitions Result Set

The Definitions result set contains information about a list of items that contains the definition for *@Term*. Each row in the result set contains the document identifier of an item and the corresponding definition related information. The result set MUST contain zero to 10 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(2000),
{title}        nvarchar(2000);

```

DocId: The identifier of an item.

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

Sentence: The extracted definition for the specified term.

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.

3.1.4.6.2 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. Each row in the result set contains the document identifier, property identifier, and the corresponding value of the managed property for the item. The result set contains zero rows if there are no high confidence results for the specified *@Term*.

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

```
Docid          int,  
pid            int,  
strVal        nvarchar(64),  
binVal        image,  
length        int;
```

Docid: The identifier of an item for which at least one high confidence property is equal to *@Term* or whose author managed property is equal to *@Author*.

pid: The identifier of a high confidence property.

strVal: The value of the high confidence property. If the value of the property is greater than the size of **strVal**, then the overflow is stored in **binVal**.

binVal: A binary image that contains the value of the part of high confidence property that did not fit into **strVal**. MUST be NULL if the value of the of high confidence property fits into **strVal**.

length: The length of **binVal**. It MUST be set to zero if **binVal** is NULL.

3.1.4.6.3 Search Property Result Set

The Search Property result set contains document identifiers and managed property values for a list of items stored in the temporary table **#Temp** (section [2.2.5.7](#)) or specified in SQL statement *@SqlSelect*.

The result set contains zero or more rows per item. For each item, the result MUST contain, if available, values for all managed properties specified in *@SqlSelect*. Each row in the result set contains the document identifier, 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 [2.2.5.1](#) for more details.

The temporary table **#Temp** (section [2.2.5.7](#)) MUST be truncated after the stored procedure is executed. The result set MUST contain zero or more rows.

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

```
DocId          int,  
Pid            int,  
llVal         bigint,  
strVal        nvarchar(64),  
binVal        image,  
{length}      int,  
fltVal        float;
```

DocId: The identifier of an item.

Pid: The identifier of a managed property.

IVal: The numeric value of the managed property. It MUST be NULL if the variant type of the managed property is not **integer**, **Boolean**, or **string**. It holds a hash of the **strVal** column, if **strVal** is not NULL.

strVal: The string value of the managed property. If the value is greater than the size of **strVal**, then the overflow is stored in **binVal**. It MUST be NULL if the managed property with the **Pid** is not of type **string**.

binVal: A binary image that contains the overflow from **strVal**. It MUST be NULL if the managed property value fits into **strVal**.

{length}: The length of the **binVal** binary image. It MUST be zero if **binVal** is NULL.

fltVal: The floating-point numeric value of the managed property. It MUST be NULL if the managed property is not of type **floating-point**.

3.1.4.7 **proc_MSS_GetSchemaHighLevelInfo**

For details see [\[MS-SQLPGAT\]](#).

3.1.4.8 **proc_MSS_GetSchemaParameters**

For details see [\[MS-SQLPGAT\]](#).

3.1.4.9 **proc_MSS_GetVectorI4ConfigurationProperty**

The **proc_MSS_GetVectorI4ConfigurationProperty** stored procedure is called to retrieve the integer value of a vector configuration property from the search shared application object or the configuration property structure and return it in the result set. If the property is found on the search shared application object this value MUST be returned; if the property is found in the configuration property structure, this value MUST be returned. Otherwise **NULL MUST be returned**.

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

```
PROCEDURE proc_MSS_GetVectorI4ConfigurationProperty (  
    @Name          nvarchar(64)  
);
```

@Name: The name of the vector configuration property.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.9.1 **Vector Configuration Property Result Set**

The Vector Configuration Property result set returns a list of the property values. The result set MUST contain zero or more rows, each corresponding to the value of a vector configuration property, if the property is an integer type.

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

```
{VectorValue}          int;
```

{VectorValue}: The value of the property.

3.1.4.10 **proc_MSS_QLog_AllocateClickIds**

The **proc_MSS_QLog_AllocateClickIds** stored procedure is called to identify a list of click identifiers that can be used to uniquely log search query information. See **Query Logging Operations** as specified in section [1.3.2](#).

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 the calling procedure MUST allocate to the log information about a search query, starting from the value returned in the result set when this stored procedure is executed.

Return Code Values: An integer which MUST be 0.

Results Sets: MUST NOT return any result set if *@numIds* is NULL. Otherwise, MUST return the following result set:

3.1.4.10.1 **Click Id Result Set**

The Click Id result set returns a 64-bit number that MUST be used as the click identifier in the **proc_MSS_GetManagedPropertyAliases** (section [3.3.4.4](#)) stored procedure. 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 calling protocol client can allocate any number 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 previous invocations.

3.1.4.11 **proc_MSS_QLog_DeleteOld**

The **proc_MSS_QLog_DeleteOld** stored procedure is called to delete search query log information from the **Historical Query Log Set** (section [3.1.1.8](#)), **Detailed Query Log Set** (section [3.1.1.9](#)) and **Summary Query Log Set** (section [3.1.1.10](#)) that are logged before a certain time period as specified in the following paragraph.

All the data in the **Historical Query Log Set** (section [3.1.1.8](#)) older than 397 days old MUST be deleted. All the data in the **Detailed Query Log Set** (section [3.1.1.9](#)) older than 366 days old

MUST get deleted. All the data in the **Summary Query Log Set** (section [3.1.1.10](#)) older than 31 days old MUST be deleted.

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

```
PROCEDURE proc_MSS_QLog_DeleteOld();
```

Return Code Values: An integer which MUST be 0.

Result Sets: SHOULD NOT [<5>](#) return any result sets. The protocol client MUST ignore any result sets returned by this stored procedure.

3.1.4.12 proc_MSS_QLog_InsertQueryInfo

The **proc_MSS_QLog_InsertQueryInfo** stored procedure is called to process the **Raw Query Log set** (section [3.1.1.7](#)) information from the **MSSQLLogUnprocessed** (section [2.2.5.4](#)) and **MSSQLLogNonClickedUnprocessed** (section [2.2.5.5](#)) tables and populate the **Historical Query Log Set** (section [3.1.1.8](#)), **Detailed Query Log Set** (section [3.1.1.9](#)), and **Summary Query Log Set** (section [3.1.1.10](#)). This stored procedure MUST truncate all the information from **MSSQLLogUnprocessed** (section [2.2.5.4](#)) and **MSSQLLogNonClickedUnprocessed** (section [2.2.5.5](#)) 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 [<6>](#) return any result sets. The protocol client MUST ignore any result sets returned by this stored procedure.

3.1.4.13 proc_MSS_QLog_LowClickThrough

The **proc_MSS_QLog_LowClickThrough** stored procedure is called to retrieve the top 300 logged search queries with the lowest clickthrough rate in the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_LowClickThrough (  
    @siteGUID                uniqueidentifier,  
    @isSspLevel              bit,  
    @showZeros                bit  
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the clickthrough rate is needed. This MUST be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This MUST be an **IsSSPLevel Flag** Data Type, as specified in section [2.2.1.8](#).

@showZeros: A bit flag that specifies whether the result set contain logged search queries with zero clickthrough rate or not. This parameter MUST be set to a value listed in the following table.

Value	Description
0	The result set MUST NOT contain logged search queries with zero clickthrough rate
1	The result set MUST contain logged search queries with zero clickthrough rate

Return Code Values: An integer that MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.13.1 Low ClickThrough Result Set

The Low Clickthrough Result Set contains a list of logged search queries and the corresponding clickthrough rate. The result set MUST contain zero or more rows, each corresponding to a logged search query. The result set MUST NOT have more than 300 rows. The rows in the result set MUST be sorted in the ascending order of **{clickRatio}** and then the rows with the same value for **{clickRatio}** MUST be sorted in the descending order of **{numQueries}**.

The result set MUST contain zero rows if *@isSspLevel* is set to 0 and the *@siteGUID* does not correspond to a site (or site collection).

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

```

queryString          nvarchar(1024),
scope                nvarchar(450),
numScopes            int,
{numQueries}         int,
{clickRatio}         decimal,
resultsUrl           nvarchar(1024),
contextualScope      nvarchar(100),
contextualScopeUrl  nvarchar(1024);

```

queryString: The query text that was executed.

scope: The search scopes under which this search query was executed. The value MUST be a Scopes data type, as specified in **Scope** (section [2.2.1.3](#)).

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.

{numQueries}: The total number of times the logged search query was executed in the past 30 days.

{clickRatio}: The **click ratio** for this logged search query

resultsUrl: The URI of the search results page in which the search results were displayed for the search query.

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

contextualScopeUrl: The URI of **contextualScope**.

3.1.4.14 proc_MSS_QLog_QueriesByDay

The **proc_MSS_QLog_QueriesByDay** stored procedure is called to get the number of search queries issued per day for the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_QueriesByDay (
    @siteGUID          uniqueidentifier,
    @isSspLevel       bit
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the number of search queries issued per day is needed. This MUST be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This MUST be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.14.1 Queries By Day Result Set

The Queries By Day result set contains a list of dates and the corresponding number of search queries issued on that date. This result set MUST contain zero or more rows, each corresponding to a single day. A day MUST not be returned if there were no search queries executed during that day. This result set MUST NOT have more than 30 rows. The rows in the result set MUST be sorted in the ascending order, according to **searchDate**. The result set MUST contain zero rows if *@isSspLevel* is set to 0 and the *@siteGUID* does not correspond to a site (2) (or site collection).

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

```
{numQueries}          int,
searchDate             smalldatetime;
```

{numQueries}: The total number of search queries executed during the **searchDate**.

searchDate: A date during which one or more search queries were executed.

3.1.4.15 proc_MSS_QLog_QueriesByMonth

The **proc_MSS_QLog_QueriesByMonth** stored procedure is called to get the number of search queries issued per month for the past 12 months, including the current month.

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

```
PROCEDURE proc_MSS_QLog_QueriesByMonth (
    @siteGUID          uniqueidentifier,
    @isSspLevel       bit
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the number of search queries issued per month is needed. This MUST be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This **MUST** be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.15.1 Queries By Month Result Set

The Queries By Month result set contains a list of months and the corresponding number of search queries executed on that month. This result set MUST contain zero or more rows, each corresponding to a month. A month MUST not be returned if there were no search queries executed during that month. It MUST NOT have more than 12 rows. The rows in the result set MUST be sorted in the ascending order, according to **searchMonth**. The result set MUST contain zero rows if **@isSspLevel** is set to 0 and the **@siteGUID** does not correspond to a site (2) (or site collection).

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

```
searchMonth          smalldatetime,  
{numQueries}        int;
```

searchMonth: A datetime that represents the 1st day of a month within the past year, including the current month, during which one or more search queries were executed.

{numQueries}: The total number of search queries executed during **searchMonth**.

3.1.4.16 proc_MSS_QLog_QueriesByScope

The **proc_MSS_QLog_QueriesByScope** stored procedure is called to get the total number of search queries issued under every search scope in the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_QueriesByScope (  
    @siteGUID          uniqueidentifier,  
    @isSspLevel        bit  
) ;
```

@siteGUID: The GUID of the site(2) (or the site collection) for which the number of search queries issued per search scope is needed. This MUST be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This **MUST** be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.16.1 Queries By Scope Result Set

The Queries By Scope result set contains a list of search scopes and the corresponding number of search queries executed under the search scope. This result set MUST contain zero or more rows, each corresponding to a search scope. A search scope MUST NOT be returned if there were no queries executed in that search scope. The rows in the result set MUST be sorted in descending order, according to **{numQueries}**. The result set MUST contain zero rows if *@isSspLevel* is set to 0 and the *@siteGUID* does not correspond to a site (2) (or site collection).

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

```
scope                nvarchar(450),
numScopes            int,
{numQueries}         int,
contextualScope      nvarchar(100);
```

scope: The search scope under which this search query was executed. The value MUST be a Scopes data type, as specified in **Scope** (section [2.2.1.3](#)).

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.

{numQueries}: The total number of search queries executed under this search scope.

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

3.1.4.17 proc_MSS_QLog_TopBestBets

The **proc_MSS_QLog_TopBestBets** stored procedure is called to retrieve the 300 best bets with most number of clicks over the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_TopBestBets (
    @siteGUID          uniqueidentifier,
    @isSspLevel        bit
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the top best bets are needed. This MUST be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This **MUST** be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

Return Code Values: An integer which MUST be 0.

Results Sets: MUST return the following result set:

3.1.4.17.1 Top Best Bets Result Set

The Top Best Bets result set returns a list of best bets and the corresponding number of clicks on the best bets during the past 30 days, including the current day. The result set MUST contain zero or more rows, each corresponding to a best bet. The result set MUST NOT have more than 300

rows. A best bet with zero clicks **MUST** not be returned. The rows in the result set **MUST** be sorted in descending order, according to **numClicks**. The result set **MUST** contain zero rows if *@isSspLevel* is set to 0 and the *@siteGUID* does not correspond to a site (or site collection).

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

```
bestBet          nvarchar(100),
numClicks        int;
```

bestBet: The title of the best bet.

numClicks: The total number of clicks on the best bet in the past 30 days, including the current day. This **MUST** be greater than 0.

3.1.4.18 proc_MSS_QLog_TopDestinations

The **proc_MSS_QLog_TopDestinations** stored procedure is called to retrieve the URI of the relevant result URIs with most number of clicks over the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_TopDestinations (
    @siteGUID          uniqueidentifier,
    @isSspLevel        bit,
    @topResultsCount   int
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the top clicked search result URIs are needed. This **MUST** be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This **MUST** be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

@ topResultsCount: The upper limit on the number of relevant search result **URLs** that **SHOULD** be returned.

Return Code Values: An integer which **MUST** be 0.

Result Sets: **MUST** return the following result set:

3.1.4.18.1 Top Destinations Result Set

The Top Destinations result set contains a list of search result URIs and the corresponding number of clicks on each search result URI during the past 30 days, including the current day. Each row of the result set corresponds to a search result URI. The result set **MUST NOT** contain more than *@topResultsCount* number of rows. The rows in the result set **MUST** be sorted in descending order, according to **numClicks**. The result set **MUST** contain zero rows if *@isSspLevel* is set to 0 and the *@siteGUID* does not correspond to a site (2) (or site collection).

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

```
clickedUrl          nvarchar(1024),
```

```
numClicks          int;
```

clickedUrl: The search result URI that was clicked in the search results page.

numClicks: The total number of clicks on the search result URI in the past 30 days, including the current day. This MUST be greater than 0.

3.1.4.19 proc_MSS_QLog_TopQueries

The **proc_MSS_QLog_TopQueries** stored procedure is called to retrieve the logged search queries that were issued the most number of times in the past 30 days.

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

```
PROCEDURE proc_MSS_QLog_TopQueries (  
    @siteGUID          uniqueidentifier,  
    @isSspLevel        bit,  
    @topResultsCount   int  
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the top logged search queries are needed. This MUST be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This MUST be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

@topResultsCount: The upper limit on number of logged search queries that SHOULD be returned.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.19.1 Top Queries Result Set

The Top Queries result set contains a list of logged search queries and the corresponding to the number of times the logged search query was executed in the past 30 days. The result set MUST contain zero or more rows, each corresponding to a logged search query. The result set MUST contain zero rows if no queries were executed in the last 30 days. The result set MUST NOT have more than *@topResultsCount* number of rows. The rows in the result set are in descending order, according to the number of times the logged search query was issued in the past 30 days, including the current day. The result set MUST contain zero rows if *@isSspLevel* is set to 0 and the *@siteGUID* does not correspond to a site (2) (or site collection).

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

```
queryString        nvarchar(1024),  
scope              nvarchar(450),  
numScopes          int,  
{numQueries}      int,  
resultsUrl         nvarchar(1024),  
contextualScope    nvarchar(100),  
contextualScopeUrl nvarchar(1024);
```

queryString: The query text that was executed.

scope: The search scope under which this search query was executed. The value MUST be a Scopes data type, as specified in **Scope** (section [2.2.1.3](#)).

numScopes: The number of search scope names that are delimited by comma in **scope**.

{numQueries}: The total number of times the logged search query was executed in the past 30 days.

resultsUrl: The URI of the search results page in which the search results were displayed for the search query.

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

contextualScopeUrl: The URI of **contextualScope**.

3.1.4.20 **proc_MSS_QLog_TopSiteCollections**

The **proc_MSS_QLog_TopSiteCollections** stored procedure is called to retrieve the total number of search queries executed on each site collection in the current SSP over the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_TopSiteCollections (  
    @siteGUID                uniqueidentifier,  
    @isSspLevel              bit  
);
```

@siteGUID: This parameter is not used.

@isSspLevel: This parameter is not used.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.20.1 **Top Site Collections Result Set**

The Top Site Collections result set contains a list of site collections and the corresponding number of search queries that were executed from that site collection in the past 30 days. The result set MUST contain zero or more rows, each corresponding to a site collection. If there were no queries executed in any of the site collections then the result MUST contain zero rows. The rows in the result set SHOULD be sorted in descending order, according to **{numQueries}**.

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

```
siteGUID                uniqueidentifier,  
{numQueries}           int;
```

siteGUID: The GUID of the site collection for which the results will be retrieved. This MUST be a site collection identifier, as specified in **Site Collection Identifier** (section [2.2.1.1](#)).

{numQueries}: The number of search queries executed from the site collection identified by **siteGUID** over the past 30 days, including the current day.

3.1.4.21 **proc_MSS_QLog_ZeroBestBets**

The **proc_MSS_QLog_ZeroBestBets** stored procedure is called to retrieve the top 300 query texts with zero best bets in the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_ZeroBestBets (  
    @siteGUID                uniqueidentifier,  
    @isSspLevel              bit  
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the query text that returned zero best bets is needed. This **MUST** be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This **MUST** be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

Return Code Values: An integer which **MUST** be 0.

Result Sets: **MUST** return the following result set:

3.1.4.21.1 **Zero best bets Result Set**

The Zero Best Bets result set contains a list of query text and the corresponding number of times zero best bets were returned for the query text. The result set **MUST** have zero or more rows, each corresponding to a query text. The result set **MUST NOT** have more than 300 rows. The rows in the result set **MUST** be sorted in the descending order of **{numQueries}**. The result set **MUST** contain zero rows if **@isSspLevel** is set to 0 and the **@siteGUID** does not correspond to a site (or site collection).

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

```
queryString                nvarchar(1024),  
{numQueries}                int,  
resultsUrl                  nvarchar(1024);
```

queryString: The query text that returned zero best bets.

{numQueries}: The total number of times the query text was executed in the past 30 days, including the current day.

resultsUrl: The URI of the search results page in which the search results were displayed for the search query

3.1.4.22 **proc_MSS_QLog_ZeroResults**

The **proc_MSS_QLog_ZeroResults** stored procedure is called to retrieve the top 300 logged search queries with zero relevant results in the past 30 days, including the current day.

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

```
PROCEDURE proc_MSS_QLog_ZeroResults (
    @siteGUID          uniqueidentifier,
    @isSspLevel       bit
);
```

@siteGUID: The GUID of the site (2) (or the site collection) for which the logged search query that returned zero relevant results is needed. This **MUST** be a site identifier (or a site collection identifier), as specified in **Site Identifier** (section [2.2.1.2](#)) or **Site Collection Identifier** (section [2.2.1.1](#)) respectively.

@isSspLevel: A bit flag that specifies whether the result is needed for a specific site (or a site collection) or for the entire SSP. This **MUST** be an **IsSSPLevel Flag** data type, as specified in **IsSSPLevel Flag** (section [2.2.1.8](#)).

Return Code Values: An integer which **MUST** be 0.

Result Sets: **MUST** return the following result set:

3.1.4.22.1 Zero Results Result Set

The Zero Results result set contains a list of logged search queries and the corresponding number of times the logged search query returned zero relevant results. The result set **MUST** contain zero or more rows, each corresponding to a logged search query. The result set **MUST NOT** have more than 300 rows. The rows in the result set **MUST** be sorted in the descending order of **{numQueries}**. The result set **MUST** contain zero rows if *@isSspLevel* is set to 0 and *@siteGUID* does not correspond to a site (or site collection).

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

```
queryString          nvarchar(1024),
scope                nvarchar(450),
numScopes            int,
{numQueries}         int,
resultsUrl           nvarchar(1024),
contextualScope      nvarchar(100),
contextualScopeUrl  nvarchar(1024);
```

queryString: The query text that returned zero results.

scope: The search scopes under which this search query was executed. The value **MUST** be a Scopes data type, as specified in **Scope** (section [2.2.1.3](#)).

numScopes: The number of search scope names that are delimited by comma in **scope**.

{numQueries}: The total number of times the query text was executed in the past 30 days, including the current day.

resultsUrl: The URI of the search results page in which the search results were displayed for the search query.

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

contextualScopeUrl: The URI of contextualScope.

3.1.4.23 `proc_MSS_RetrieveDuplicateHashes`

The `proc_MSS_RetrieveDuplicateHashes` stored procedure is called to retrieve identifiers for a list of items. The identifiers are used for duplicate result removal. The document identifiers for the list of items MUST be stored in the temporary table `#Temp` (section [2.2.5.7](#)). If two items have the same identifier, then there is a high probability that they are similar.

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

```
PROCEDURE proc_MSS_RetrieveDuplicateHashes();
```

Return Code Values: An integer which MUST be 0.

Results Sets: MUST return the following result set:

3.1.4.23.1 Duplicate Hashes Result Set

The Duplicate Hashes result set contains information about identifiers for a list of items. Each row in the result set contains document identifier of an item and the corresponding identifiers that are used for duplicate result removal. The result set MUST contain zero or 6 rows per item. The rows in the result set MUST be in ascending order according to **value**, as specified in the temporary table `#Temp` (section [2.2.5.7](#)).

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

```
DocId                int NOT NULL,  
HashVal              bigint;
```

DocId: The unique identifier of an item.

HashVal: One of the identifiers of the item used for duplicate result removal.

3.1.4.24 `proc_MSS_RetrieveDuplicateHashesEx`

The `proc_MSS_RetrieveDuplicateHashesEx` stored procedure is called to retrieve identifiers for a list of items. The identifiers are used for duplicate result removal. If two items have the same identifier, then there is a high probability that they are similar.

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

```
PROCEDURE proc_MSS_RetrieveDuplicateHashesEx (  
    @docids                nvarchar(2048)  
);
```

@docids: A string that contains document identifiers for a list of items. The document identifiers are separated by commas, with whitespace allowed between commas and document identifiers.

Return Code Values: An integer which MUST be 0.

Results Sets: MUST return the following result set:

3.1.4.24.1 Duplicate HashesEx Result Set

The Duplicate HashesEx result set contains information about identifiers for a list of items. Each row in the result set contains the document identifier of an item and the corresponding identifiers used for duplicate result removal. The result set MUST contain zero or 6 rows per item.

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

```
DocId                int NOT NULL,  
HashVal              bigint;
```

DocId: The unique identifier of an item.

HashVal: One of the identifiers of the item used for duplicate result removal.

3.1.4.25 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 as specified in [\[MS-SQLPADM\]](#) section 2.2.2.15.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following two result sets:

3.1.4.25.1 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 zero or more rows, each row corresponding to a single search scope consumer

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

```
ConsumerName        nvarchar(60);
```

ConsumerName: The name of the search scope consumer.

3.1.4.25.2 Last Consumer Change ID Result Set

The Last Consumer Change ID 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;
```

LastConsumerChangeID: The version whose value MUST be a LastConsumerChangeID data type as specified in [\[MS-SQLPADM\]](#) section 2.2.2.13.

3.1.4.26 `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: An integer which MUST be 0.

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

3.1.4.26.1 **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,  
Name             nvarchar(60),  
Description      nvarchar(300),  
DisplayInAdminUI bit,  
AlternateResultsPageUrl nvarchar(2048),  
CompilationState smallint;
```

ScopeID: Unique identifier of the search scope.

Name: The name of the search scope.

Description: The description of the search scope.

DisplayInAdminUI: `<7>` If set to 1, the search scope is displayed in the administration user interface. Otherwise, it MUST be set to 0.

AlternateResultsPageUrl: The URI of an alternate search results page to display the results of a search performed on this search scope. This value can be set to NULL.

CompilationState: The **search scope compilation** state of the given search scope. The value MUST be a **CompilationState** data type as specified in [\[MS-SQLPADM\]](#) section 2.2.2.4.

3.1.4.26.2 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,  
Name                nvarchar(60),  
DefaultScopeID     int;
```

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.

3.1.4.26.3 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,  
ScopeID             int,  
Rank                int;
```

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 the position at which this search scope display group **MUST** be displayed to the administrator.

3.1.4.26.4 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;
```

LastUpdate: The last update value of the search scope consumer. The value **MUST** be a **LastUpdate** data type as specified in [\[MS-SQLPDM\]](#) section 2.2.2.15.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details for Windows Sharepoint Server

The client role that is served by the Microsoft Office SharePoint Server 2007 role is described in this section. The client role can make requests for search query execution, managed properties retrieval operations, and search query log operations.

A user sends a request to the protocol client (front-end Web server) to perform a search query. In this case, the front-end Web server performs a list of tasks necessary to return items which match this search query. The tasks include retrieving items from a back-end database server, eliminating certain items based on security information or because of duplicate results, and retrieving managed properties for a list of indexed documents. The task of retrieving an item from a back-end database server **MUST** be accomplished by querying the MSSDocProps table. The tasks of item elimination **MUST** be accomplished by executing the **proc_MSS_FetchSDs**, **proc_MSS_RetrieveDuplicateHashes**, and **proc_MSS_RetrieveDuplicateHashesEx** stored procedures. The task of retrieving managed properties from a back-end database server **MUST** be accomplished by querying the MSSDocProps table.

A client (such as a user) can also send a request to the protocol client (front-end Web server) to log a search query or return a search query log report. In this case, the front-end Web server calls the appropriate 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 client (front-end Web server) 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 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:

- Document identifier

3.2.1.2 Result Document Property Set

The Result Document Property set contains a list of items that are part of a search result set with zero or more properties. The attributes of an item property set are:

- Document identifier

- Property identifier
- Property value

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

3.3 Windows SharePoint Services Server Details

The Windows Sharepoint Services server role is described in this section. The Windows SharePoint Services role only 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 back-end database server maintains the following set of data for this protocol within the metadata schema. Data within the back-end database server are maintained until updated or removed. The following objects are defined here:

- Managed property
- Search security descriptor
- Item Identifier set used for duplicate result removal

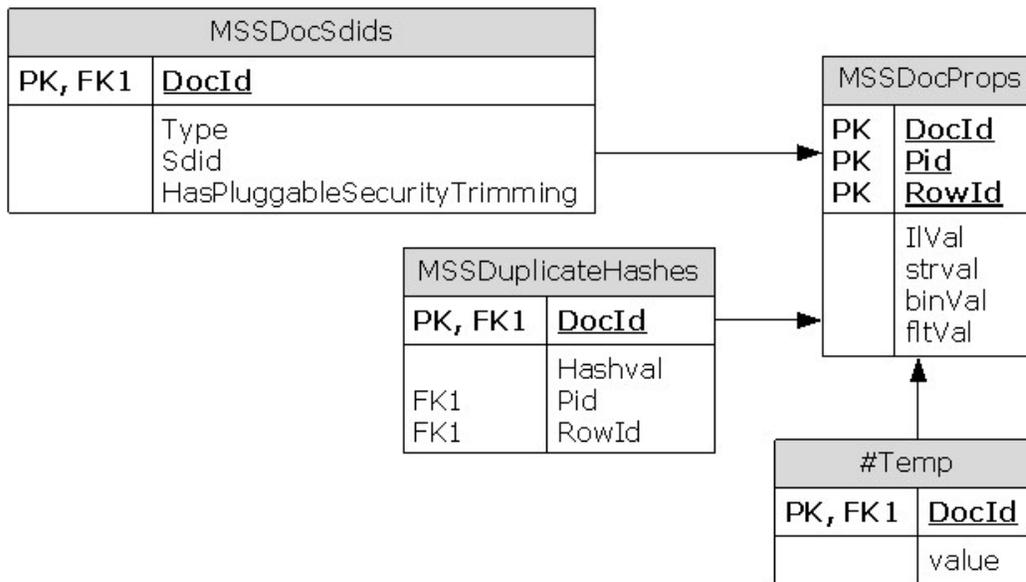


Figure 5: Query Execution Abstract Data Model

The following subsections define each of these data structures.

3.3.1.1 Managed property value set

See **Managed Property Value Set** (section [3.1.1.1](#)).

3.3.1.2 Search security descriptor set

See **Search Security Descriptor Set** (section [3.1.1.2](#)).

3.3.1.3 Item identifier set

See **Item Identifier Set** (section [3.1.1.3](#)).

3.3.2 Timers

An execution timeout timer on the protocol server governs the execution time for the client's requests. The amount of time is specified by a timeout value that is configured on the protocol server for all connections.

3.3.3 Initialization

Listening endpoints are set up on the back-end database server to handle inbound TDS requests.

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

The data structures, stored procedures, and actual data are persisted by the back-end database server within databases, so any operations to initialize the state of the database **MUST** occur before the back-end database server can use this protocol.

3.3.4 Message Processing Events and Sequencing Rules

Unless otherwise specified, all stored procedures defined in this section are located in the search 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}*.

Unless otherwise specified, all stored procedure input parameters **MUST NOT** be NULL. As stored procedures use the input parameters for data retrieval from tables, failure to provide valid values will (unless otherwise specified) cause an error as specified in [\[MS-TDS\]](#) that **MUST** be handled appropriately by the protocol client or the system behavior is indeterminate.

The following table summarizes the stored procedures available in Windows SharePoint Services search application for search query execution and managed properties retrieval operations that are defined in this specification:

Procedure Name	Description
proc_MSS_FetchSDs (specified in section 3.3.4.1)	Retrieves search security descriptors for the specified list of items.
proc_MSS_GetConfigurationProperty (specified in [MS-SQLPGAT])	Gets the value of a property of the configuration property structure.
proc_MSS_GetManagedProperties (specified in [MS-SQLPGAT])	Lists managed properties from the metadata schema which were added or modified on or after the given time.
proc_MSS_GetManagedPropertyAliases (specified in section 3.1.4.4)	Lists the aliases for managed properties from the metadata schema which were added or modified on or after the given time..
proc_MSS_GetMultipleResults	Retrieves values for specified managed properties for the specified list of items.
proc_MSS_GetSchemaHighLevelInfo (specified in [MS-SQLPGAT])	Retrieves last modified and last deleted timestamps from the metadata schema .
proc_MSS_GetSchemaParameters (specified in [MS-SQLPGAT])	Retrieves a list of schema parameters from the metadata schema.
proc_MSS_GetVectorI4ConfigurationProperty	Gets the value vector configuration property structure from the configuration property structure.
proc_MSS_RetrieveDuplicateHashes (specified in section 3.1.4.23)	Retrieves identifiers for the list of items specified in the temporary table #Temp (section 2.2.5.7). The identifiers are used for duplicate results removal.
proc_MSS_RetrieveDuplicateHashesEx (specified in section 3.1.4.24)	Retrieves identifiers for the specified list of items. The identifiers are used for duplicate

Procedure Name	Description
	results removal.

3.3.4.1 proc_MSS_FetchSDs

For details see section [3.3.4.1](#).

3.3.4.2 proc_MSS_GetConfigurationProperty

For details see [\[MS-SQLPGAT\]](#).

3.3.4.3 proc_MSS_GetManagedProperties

For details see [\[MS-SQLPGAT\]](#).

3.3.4.4 proc_MSS_GetManagedPropertyAliases

For details see section [3.1.4.4](#).

3.3.4.5 proc_MSS_GetMultipleResults

The **proc_MSS_GetMultipleResults** stored procedure is called to retrieve managed properties for a list of items. The list of managed properties is specified in a SQL statement. The list of items is stored in the temporary table **#Temp** (section [2.2.5.7](#)). If the temporary table is not available then the list of documents **MUST** be specified in *@SqlSelect*.

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

```
PROCEDURE proc_MSS_GetMultipleResults(
    @RequestTypes      int,
    @Term              nvARCHAR(100),
    @Author            nvARCHAR(100),
    @SqlSelect         nvARCHAR(2048)
);
```

@RequestTypes: The type of result expected. The only possible value **MUST** be:

Value	Description
0x00000001	Managed properties for a list of items MUST be returned.

@Term: This parameter is not used.

@Author: All the restrictions imposed on author managed property.

@SqlSelect: The T-SQL statement that **MUST** be executed to get the managed properties for a list of items. The document identifiers of the items for which property values are to be returned **MUST** be stored in the **docid** field of the temporary table **#Temp** (section [2.2.5.7](#)). The T-SQL statement **MUST** retrieve data from the **MSSDocProps** (section [2.2.5.1](#)), **MSSDocSdids** (section [2.2.5.2](#)) and **#Temp** (section [2.2.5.7](#)) tables

Return Code Values: An integer which **MUST** be 0.

Result Sets: MUST return the following result set:

3.3.4.5.1 Search Property Result Set

The Search Property result set contains document identifiers and managed property values for a list of items stored in the temporary table **#Temp** (section [2.2.5.7](#)) or specified in SQL statement *@SqlSelect*.

The result set contains zero or more rows per item. For each item, the result MUST contain values for all managed properties specified in *@Sqlselect* if available. Each row in the result set contains a document identifier, managed 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 [2.2.5.1](#) for more details.

The temporary table **#Temp** (section [2.2.5.7](#)) table MUST be truncated after the stored procedure is executed. The result set MUST contain zero or more rows.

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

```
DocId          int,  
Pid            int,  
IIVal         bigint,  
strVal        nvarchar(64),  
binVal        image,  
length        int,  
fltVal        float;
```

DocId: The identifier of an item.

Pid: The identifier of the managed property.

IIVal: The numeric value of the managed property. It MUST be NULL if the variant type of the managed property is not integer, Boolean, or string. It holds a hash of the **strVal** column, if **strVal** is not NULL.

strVal: The string value of the managed property. If the value is greater than the size of **strVal**, then the overflow is stored in **binVal**. It MUST be NULL if the managed property with **Pid** is not of type string.

binVal: A binary image that contains the overflow from **strVal**.

length: The length of the **binVal** binary image. It MUST be zero if **binVal** is NULL.

fltVal: The floating-point numeric value of the managed property. It MUST be NULL if the managed property with **Pid** is not of type floating-point.

3.3.4.6 proc_MSS_GetSchemaHighLevelInfo

For details see [\[MS-SQLPGAT\]](#).

3.3.4.7 proc_MSS_GetSchemaParameters

For details see [\[MS-SQLPGAT\]](#).

3.3.4.8 **proc_MSS_GetVectorI4ConfigurationProperty**

The **proc_MSS_GetVectorI4ConfigurationProperty** stored procedure is called to retrieve the integer value of a vector configuration property from the configuration property structure and return it in the result set. If the property is found in the configuration property structure, this value **MUST** be returned. Otherwise **NULL MUST be returned**.

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

```
PROCEDURE proc_MSS_GetVectorI4ConfigurationProperty (
    @Name          nvarchar(64)
);
```

@Name: The name of the vector configuration property.

Return Code Values: An integer which **MUST** be 0.

Result Sets: **MUST** return the following result set:

3.3.4.8.1 **Vector Configuration Property Result Set**

The Vector Configuration Property result set returns a list of the property values. The result set **MUST** contain zero or more rows, each corresponding to the value of a vector configuration property, if the property is an integer type.

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

```
{VectorValue}          int;
```

{VectorValue}: The value of the property.

3.3.4.9 **proc_MSS_RetrieveDuplicateHashes**

For details see section **proc_MSS_RetrieveDuplicateHashes** (section [3.1.4.23](#)).

3.3.4.10 **proc_MSS_RetrieveDuplicateHashesEx**

For details see section **proc_MSS_RetrieveDuplicateHashesEx** (section [3.1.4.24](#)).

3.3.5 **Timer Events**

None.

3.3.6 **Other Local Events**

None.

3.4 **Protocol Client (front-end web server) details for Windows SharePoint Services Server**

The client role being served by Windows SharePoint Services server role is described in this section. The client role can submit only requests for search query execution and managed properties retrieval operations.

A user or an application sends a request to the protocol client (front-end Web server) to perform a search query. In this case, the front-end Web server performs a list of tasks necessary to return items that match this search query. The tasks include retrieving item from a back-end database server, eliminating certain items based on security information or duplicate result removal, and retrieving managed properties for the list of items. The task of retrieving an item from a back-end database server MUST be accomplished by querying the **MSSDocProps** (section [2.2.5.1](#)) table. The tasks of item elimination MUST be accomplished by executing the **proc_MSS_FetchSDs** (section [3.3.4.1](#)), **proc_MSS_RetrieveDuplicateHashes** (section [3.1.4.23](#)) and **proc_MSS_RetrieveDuplicateHashesEx** (section [3.1.4.24](#)) stored procedures. The task of retrieving managed properties from a back-end database server] MUST be accomplished by querying the **MSSDocProps** (section [2.2.5.1](#)) table.

3.4.1 Abstract Data Model

See **Abstract Data Model** (section [3.2.1](#)) of this document.

3.4.2 Timers

None.

3.4.3 Initialization

None.

3.4.4 Message Processing Events and Sequencing Rules

None.

3.4.5 Timer Events

None.

3.4.6 Other Local Events

None.

4 Protocol Examples

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 access rights to the databases on the back end database server (back-end database server), which is negotiated as part of the Tabular Data Stream ([\[MS-TDS\]](#)) protocol.

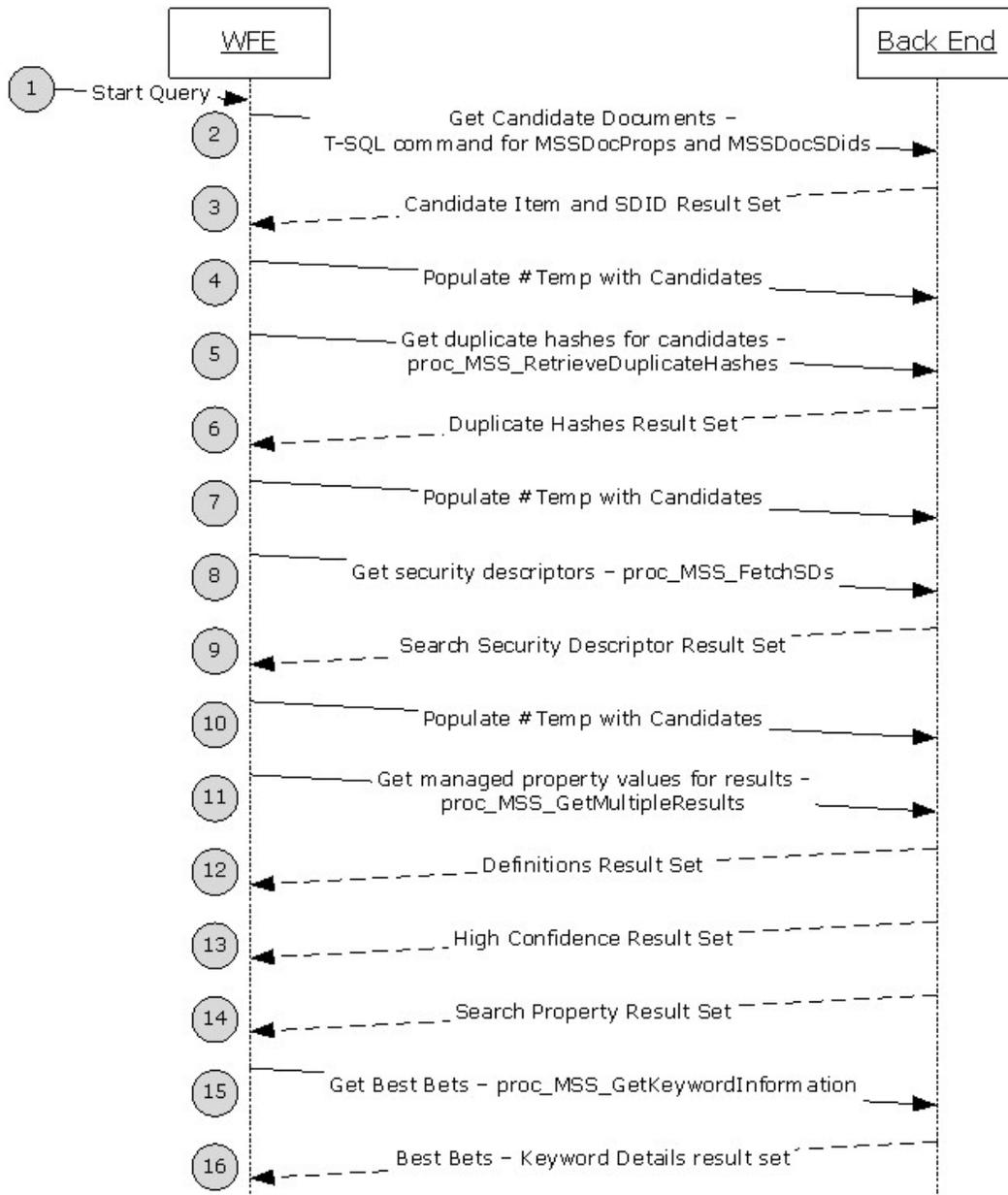


Figure 6: Query Execution

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 front-end Web server receives a query request from the user interface or Web Service.
2. The front-end Web server translates the query into a T-SQL command and sends it to the back end database server (back-end database server).
3. The back-end database server returns a set of item identifiers and search security descriptor identifiers which match the T-SQL command.
4. The front-end Web server populates the **#Temp** (section [2.2.5.7](#)) table with the set of item identifiers which are candidate query results.
5. The front-end Web server calls **proc_MSS_RetrieveDuplicateHashes** to determine which documents listed in **#Temp** (section [2.2.5.7](#)) are similar to each other.
6. The back-end database server returns a Duplicate Hashes Result Set which contains document identifiers for each document which are the same when two documents have similar content and metadata.
7. The front-end Web server populates the **#Temp** (section [2.2.5.7](#)) table with the set of item identifiers which are candidate query results.
8. The front-end Web server calls **proc_MSS_FetchSDs** to retrieve the search security descriptors that correspond to the candidate items in **#Temp** (section [2.2.5.7](#)).
9. The back-end database server returns a Search Security Descriptors Result set for the items specified in **#Temp** (section [2.2.5.7](#)).
10. The front-end Web server populates the **#Temp** (section [2.2.5.7](#)) table with the set of item identifiers which are candidate query results.
11. The front-end Web server 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 front-end Web server 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.
12. The back-end database server returns a Definitions Result set for the documents specified in *@Term*.
13. The back-end database server returns a High Confidence Result set for the documents specified in *@Term* and *@Author*.
14. The back-end database server returns a Search Property Result set as per the **@SqlSelect** statement.
15. The front-end Web server calls **proc_MSS_GetKeywordInformation** to retrieve best bet information, specifying the query term in the *@Term* parameter.
16. The back-end database server returns a Keyword Details result set for the specified term.

4.2 Query Logging and Usage Reporting

This example describes the requests made and responses returned during operations related to the search query log and usage reporting.

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

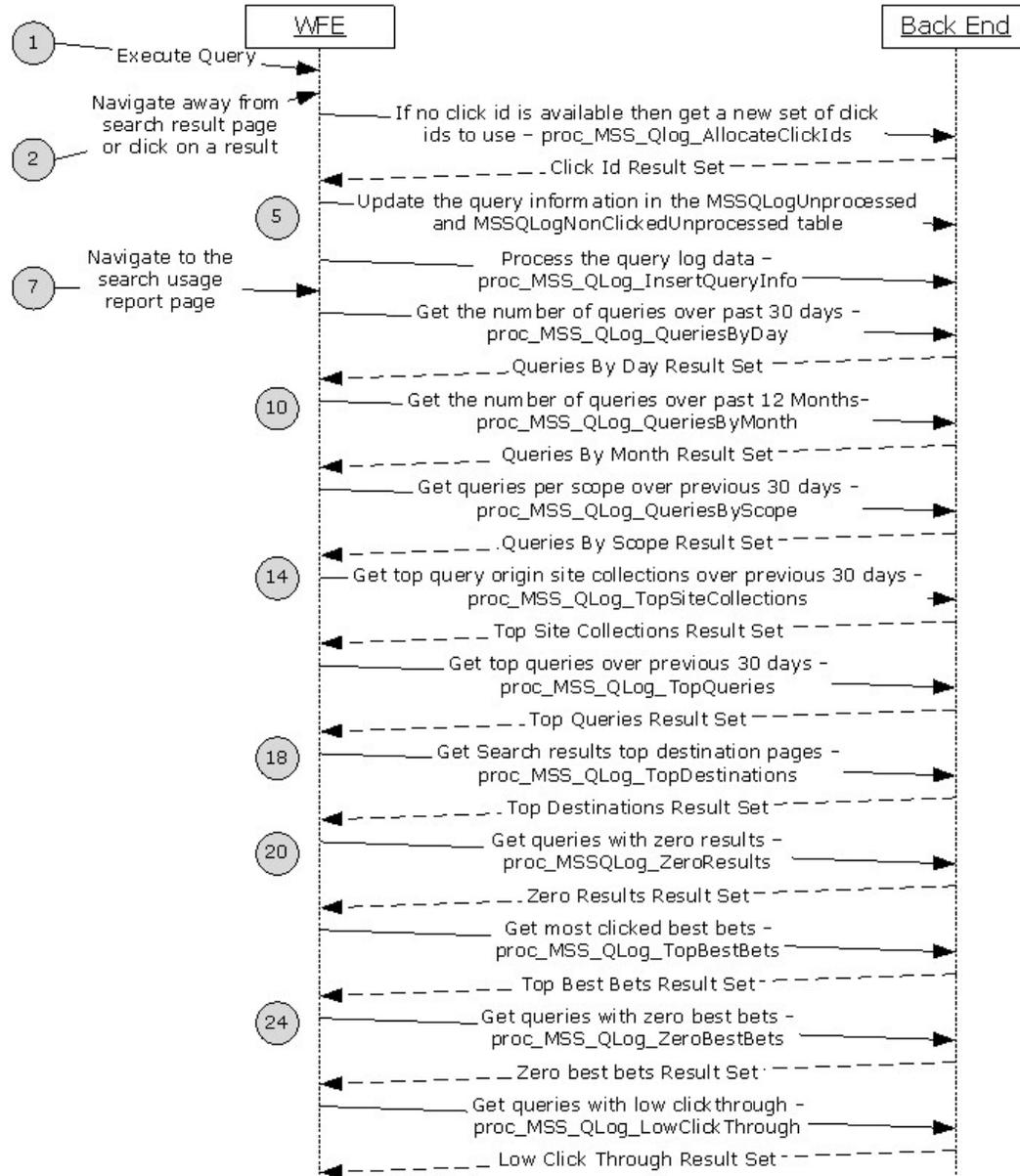


Figure 7: Query Logging and Usage Reporting

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 front-end Web server receives a query request from the user interface.
2. The user navigates away from the search results page or clicks on one of the search results.
3. The front-end Web server 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 back-end database server.
4. The back-end database server returns a click id result set.
5. The front-end Web server updates the search query log tables MSSQLLogUnprocessed and MSSQLLogNonClickedUnprocessed.
6. The front-end Web server periodically calls the **proc_MSS_QLog_InsertQueryInfo** to process the information in the MSSQLLogUnprocessed and MSSQLLogNonClickedUnprocessed tables.
7. The search administrator navigates to the search usage reports page.
8. The front-end Web server calls **proc_MSS_QLog_QueriesByDay** to retrieve the number of queries over past 30 days.
9. The back-end database server returns a Queries By Day Result set.
10. The front-end Web server calls **proc_MSS_QLog_QueriesByMonth** to retrieve the number of queries over past 12 months.
11. The back-end database server returns a Queries By Month Result set.
12. The front-end Web server calls **proc_MSS_QLog_QueriesByScope** to retrieve the number of queries per scope over past 30 days.
13. The back-end database server returns a Queries By Scope Result set.
14. The front-end Web server calls **proc_MSS_QLog_TopSiteCollections** to retrieve the top query origin site collections over previous 30 days.
15. The back-end database server returns a Top Site Collections Result set.
16. The front-end Web server calls **proc_MSS_QLog_TopQueries** to retrieve the top query over past 30 days.
17. The back-end database server returns a Top Queries Result set.
18. The front-end Web server calls **proc_MSS_QLog_TopDestinations** to retrieve the search results top destination pages.
19. The back-end database server returns a Top Destinations Result set.
20. The front-end Web server calls **proc_MSS_QLog_ZeroResults** to retrieve the search results top destination pages.
21. The back-end database server returns a Zero Results Result set.

22. The front-end Web server calls **proc_MSS_QLog_TopBestBets** to retrieve the search results top destination pages.
23. The back-end database server returns a Top Best Bets Result set.
24. The front-end Web server calls **proc_MSS_QLog_ZeroBestBets** to retrieve the search results top destination pages.
25. The back-end database server returns a Zero Best Bets Result set.
26. The front-end Web server calls **proc_MSS_QLog_LowClickThrough** to retrieve the search results top destination pages.
27. The back-end database server returns a LowClickThrough Result set.

5 Security

5.1 Security Considerations for Implementers

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

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

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

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

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

[<1> Section 2.2.5.4:](#) This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.

[<2> Section 3.1.1.7:](#) This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.

[<3> Section 3.1.1.9:](#) This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.

[<4> Section 3.1.1.10:](#) This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.

[<5> Section 3.1.4.11:](#) If a given stored procedure does an INSERT, UPDATE, or DELETE SQL operation in the database, the stored procedure returns one or more extra result sets that contain the number of records affected by the operation.

[<6> Section 3.1.4.12:](#) If a given stored procedure does an INSERT, UPDATE, or DELETE SQL operation in the database, the stored procedure returns one or more extra result sets that contain the number of records affected by the operation.

[<7> Section 3.1.4.26.1:](#) This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.

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

[#Temp table structure](#) 19

A

Abstract data model
[Best Bet Set](#) 23
client ([section 3.2.1](#) 49, [section 3.4.1](#) 56)
[Detailed Query Log Set](#) 24
[Extracted Definition Set](#) 23
[Historical Query Log Set](#) 24
Item Identifier Set ([section 3.1.1.3](#) 23, [section 3.3.1.3](#) 51)
Managed Property Value Set ([section 3.1.1.1](#) 22, [section 3.3.1.1](#) 51)
[Query Result Candidate Document Site](#) 49
[Raw Query Log Set](#) 24
[Result Document Property Set](#) 49
[Scopes](#) 23
Search Security Descriptor Set ([section 3.1.1.2](#) 22, [section 3.3.1.2](#) 51)
server ([section 3.1.1](#) 21, [section 3.3.1](#) 50)
[Summary Query Log Set](#) 25
Advanced Search Flag simple type ([section 2.2.1.4](#) 14, [section 2.2.1.4](#) 14)
[Applicability](#) 13

B

Best Bet Set
[abstract data model](#) 23
[Binary structures - overview](#) 15
[Bit fields - overview](#) 15

C

[Capability negotiation](#) 13
[Change tracking](#) 64
Client
 abstract data model ([section 3.2.1](#) 49, [section 3.4.1](#) 56)
 initialization ([section 3.2.3](#) 50, [section 3.4.3](#) 56)
 local events ([section 3.2.6](#) 50, [section 3.4.6](#) 56)
 message processing ([section 3.2.4](#) 50, [section 3.4.4](#) 56)
 [Microsoft Office SharePoint Server 2007 interface](#) 49
 [overview – Microsoft Office SharePoint Server 2007](#) 49
 [overview – Windows SharePoint Services](#) 55
 sequencing rules ([section 3.2.4](#) 50, [section 3.4.4](#) 56)
 timer events ([section 3.2.5](#) 50, [section 3.4.5](#) 56)
 timers ([section 3.2.2](#) 50, [section 3.4.2](#) 56)
 [Windows SharePoint Services interface](#) 55
Common data types
[overview](#) 14
Continued Flag simple type ([section 2.2.1.5](#) 14, [section 2.2.1.5](#) 14)

D

Data model - abstract
[Best Bet Set](#) 23
client ([section 3.2.1](#) 49, [section 3.4.1](#) 56)
[Detailed Query Log Set](#) 24
[Extracted Definition Set](#) 23
[Historical Query Log Set](#) 24
Item Identifier Set ([section 3.1.1.3](#) 23, [section 3.3.1.3](#) 51)
Managed Property Value Set ([section 3.1.1.1](#) 22, [section 3.3.1.1](#) 51)
[Query Result Candidate Document Site](#) 49
[Raw Query Log Set](#) 24
[Result Document Property Set](#) 49
[Scopes](#) 23
Search Security Descriptor Set ([section 3.1.1.2](#) 22, [section 3.3.1.2](#) 51)
server ([section 3.1.1](#) 21, [section 3.3.1](#) 50)
[Summary Query Log Set](#) 25
Data types
 Advanced Search Flag simple type ([section 2.2.1.4](#) 14, [section 2.2.1.4](#) 14)
 [common](#) 14
 Continued Flag simple type ([section 2.2.1.5](#) 14, [section 2.2.1.5](#) 14)
 DidYouMean Flag simple type ([section 2.2.1.6](#) 15, [section 2.2.1.6](#) 15)
 IsSSPLLevel Flag simple type ([section 2.2.1.8](#) 15, [section 2.2.1.8](#) 15)
 ResultView Flag simple type ([section 2.2.1.7](#) 15, [section 2.2.1.7](#) 15)
 Scope simple type ([section 2.2.1.3](#) 14, [section 2.2.1.3](#) 14)
 Site Collection Identifier simple type ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
 Site Identifier simple type ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
Data types - simple
 Advanced Search Flag ([section 2.2.1.4](#) 14, [section 2.2.1.4](#) 14)
 Continued Flag ([section 2.2.1.5](#) 14, [section 2.2.1.5](#) 14)
 DidYouMean Flag ([section 2.2.1.6](#) 15, [section 2.2.1.6](#) 15)
 IsSSPLLevel Flag ([section 2.2.1.8](#) 15, [section 2.2.1.8](#) 15)
 [overview](#) 14
 ResultView Flag ([section 2.2.1.7](#) 15, [section 2.2.1.7](#) 15)
 Scope ([section 2.2.1.3](#) 14, [section 2.2.1.3](#) 14)
 Site Collection Identifier ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
 Site Identifier ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
Detailed Query Log Set
[abstract data model](#) 24

DidYouMean Flag simple type ([section 2.2.1.6](#) 15, [section 2.2.1.6](#) 15)

E

Events

- local - client ([section 3.2.6](#) 50, [section 3.4.6](#) 56)
- local - server ([section 3.1.6](#) 49, [section 3.3.6](#) 55)
- timer - client ([section 3.2.5](#) 50, [section 3.4.5](#) 56)
- timer - server ([section 3.1.5](#) 48, [section 3.3.5](#) 55)

Examples

- [query execution](#) 57
- [query logging and usage reporting](#) 59

Extracted Definition Set

- [abstract data model](#) 23

F

[Fields - vendor-extensible](#) 13

[Flag structures - overview](#) 15

G

[Glossary](#) 7

H

Historical Query Log Set

- [abstract data model](#) 24

I

[Implementer - security considerations](#) 62

[Index of security parameters](#) 62

[Informative references](#) 9

Initialization

- client ([section 3.2.3](#) 50, [section 3.4.3](#) 56)
- server ([section 3.1.3](#) 25, [section 3.3.3](#) 51)

Interfaces – client

- [Microsoft Office SharePoint Server 2007](#) 49
- [Windows SharePoint Services](#) 55

Interfaces - server

- [Microsoft Office SharePoint Server](#) 21
- [Windows SharePoint Services](#) 50

Introduction

IsSSPLLevel Flag simple type ([section 2.2.1.8](#) 15, [section 2.2.1.8](#) 15)

Item Identifier Set

- abstract data model ([section 3.1.1.3](#) 23, [section 3.3.1.3](#) 51)

L

Local events

- client ([section 3.2.6](#) 50, [section 3.4.6](#) 56)
- server ([section 3.1.6](#) 49, [section 3.3.6](#) 55)

M

Managed Property Value Set abstract data model ([section 3.1.1.1](#) 22, [section 3.3.1.1](#) 51)

Message processing

- client ([section 3.2.4](#) 50, [section 3.4.4](#) 56)
- server ([section 3.1.4](#) 26, [section 3.3.4](#) 52)

Messages

- [#Temp table structure](#) 19
- [binary structures](#) 15
- [bit fields](#) 15
- [common data types](#) 14
- [enumerations](#) 14
- [flag structures](#) 15
- [MSSDocProps table structure](#) 16
- [MSSDocSdids table structure](#) 16
- [MSSDuplicateHashes table structure](#) 17
- [MSSQLogLocation table structure](#) 19
- [MSSQLogNonClickedUnprocessed table structure](#) 19
- [MSSQLogUnprocessed table structure](#) 17
- [result sets](#) 15
- [simple data types](#) 14
- [table structures](#) 15
- [transport](#) 14
- [view structures](#) 15
- [XML structures](#) 20

Methods

- proc_MSS_FetchSDs ([section 3.1.4.1](#) 28, [section 3.3.4.1](#) 53)
- proc_MSS_GetConfigurationProperty ([section 3.1.4.2](#) 28, [section 3.3.4.2](#) 53)
- proc_MSS_GetKeywordInformation 29
- proc_MSS_GetManagedProperties ([section 3.1.4.3](#) 28, [section 3.3.4.3](#) 53)
- proc_MSS_GetManagedPropertyAliases ([section 3.1.4.4](#) 29, [section 3.3.4.4](#) 53)
- proc_MSS_GetMultipleResults ([section 3.1.4.6](#) 30, [section 3.3.4.5](#) 53)
- proc_MSS_GetSchemaHighLevelInfo ([section 3.1.4.7](#) 33, [section 3.3.4.6](#) 54)
- proc_MSS_GetSchemaParameters ([section 3.1.4.8](#) 33, [section 3.3.4.7](#) 54)
- proc_MSS_GetVectorI4ConfigurationProperty ([section 3.1.4.9](#) 33, [section 3.3.4.8](#) 55)
- [proc MSS QLog AllocateClickIds](#) 34
- [proc MSS QLog DeleteOld](#) 34
- [proc MSS QLog InsertQueryInfo](#) 35
- [proc MSS QLog LowClickThrough](#) 35
- [proc MSS QLog QueriesByDay](#) 36
- [proc MSS QLog QueriesByMonth](#) 37
- [proc MSS QLog QueriesByScope](#) 38
- [proc MSS QLog TopBestBets](#) 39
- [proc MSS QLog TopDestinations](#) 40
- [proc MSS QLog TopQueries](#) 41
- [proc MSS QLog TopSiteCollections](#) 42
- [proc MSS QLog ZeroBestBets](#) 43
- [proc MSS QLog ZeroResults](#) 43
- proc_MSS_RetrieveDuplicateHashes ([section 3.1.4.23](#) 45, [section 3.3.4.9](#) 55)
- proc_MSS_RetrieveDuplicateHashesEx ([section 3.1.4.24](#) 45, [section 3.3.4.10](#) 55)
- [proc MSS ScopeCacheGetChangedConsumers](#) 46
- [proc MSS ScopeCacheGetChanges](#) 47

[Microsoft Office SharePoint Server 2007 interface - client](#) 49
[Microsoft Office SharePoint Server interface - server](#) 21
[MSSDocProps table structure](#) 16
[MSSDocSdids table structure](#) 16
[MSSDuplicateHashes table structure](#) 17
[MSSQLogLocation table structure](#) 19
[MSSQLogNonClickedUnprocessed table structure](#) 19
[MSSQLogUnprocessed table structure](#) 17

N

[Normative references](#) 9

O

Overview

[Property Based Query Execution & Managed Properties Retrieval Operations](#) 10

[Query Logging Operations](#) 11

[Overview \(synopsis\)](#) 9

P

[Parameters - security index](#) 62

[Preconditions](#) 13

[Prerequisites](#) 13

[proc_MSS_FetchSDs method](#) ([section 3.1.4.1](#) 28, [section 3.3.4.1](#) 53)

[proc_MSS_GetConfigurationProperty method](#) ([section 3.1.4.2](#) 28, [section 3.3.4.2](#) 53)

[proc_MSS_GetKeywordInformation method](#) 29

[proc_MSS_GetManagedProperties method](#) ([section 3.1.4.3](#) 28, [section 3.3.4.3](#) 53)

[proc_MSS_GetManagedPropertyAliases method](#) ([section 3.1.4.4](#) 29, [section 3.3.4.4](#) 53)

[proc_MSS_GetMultipleResults method](#) ([section 3.1.4.6](#) 30, [section 3.3.4.5](#) 53)

[proc_MSS_GetSchemaHighLevelInfo method](#) ([section 3.1.4.7](#) 33, [section 3.3.4.6](#) 54)

[proc_MSS_GetSchemaParameters method](#) ([section 3.1.4.8](#) 33, [section 3.3.4.7](#) 54)

[proc_MSS_GetVectorI4ConfigurationProperty method](#) ([section 3.1.4.9](#) 33, [section 3.3.4.8](#) 55)

[proc_MSS_QLog_AllocateClickIds method](#) 34

[proc_MSS_QLog_DeleteOld method](#) 34

[proc_MSS_QLog_InsertQueryInfo method](#) 35

[proc_MSS_QLog_LowClickThrough method](#) 35

[proc_MSS_QLog_QueriesByDay method](#) 36

[proc_MSS_QLog_QueriesByMonth method](#) 37

[proc_MSS_QLog_QueriesByScope method](#) 38

[proc_MSS_QLog_TopBestBets method](#) 39

[proc_MSS_QLog_TopDestinations method](#) 40

[proc_MSS_QLog_TopQueries method](#) 41

[proc_MSS_QLog_TopSiteCollections method](#) 42

[proc_MSS_QLog_ZeroBestBets method](#) 43

[proc_MSS_QLog_ZeroResults method](#) 43

[proc_MSS_RetrieveDuplicateHashes method](#) ([section 3.1.4.23](#) 45, [section 3.3.4.9](#) 55)

[proc_MSS_RetrieveDuplicateHashesEx method](#) ([section 3.1.4.24](#) 45, [section 3.3.4.10](#) 55)

[proc_MSS_ScopeCacheGetChangedConsumers method](#) 46

[proc_MSS_ScopeCacheGetChanges method](#) 47

[Product behavior](#) 63

[Property Based Query Execution & Managed Properties Retrieval Operations overview](#) 10

Q

[Query execution example](#) 57

[Query logging and usage reporting example](#) 59

[Query Logging Operations overview](#) 11

[Query Result Candidate Document Site abstract data model](#) 49

R

Raw Query Log Set

[abstract data model](#) 24

[References](#) 9

[informative](#) 9

[normative](#) 9

[Relationship to other protocols](#) 12

[Result Document Property Set abstract data model](#) 49

[Result sets - overview](#) 15

ResultView Flag simple type ([section 2.2.1.7](#) 15, [section 2.2.1.7](#) 15)

S

Scope simple type ([section 2.2.1.3](#) 14, [section 2.2.1.3](#) 14)

Scopes

[abstract data model](#) 23

Search Security Descriptor Set abstract data model ([section 3.1.1.2](#) 22, [section 3.3.1.2](#) 51)

Security

[implementer considerations](#) 62

[parameter index](#) 62

Sequencing rules

client ([section 3.2.4](#) 50, [section 3.4.4](#) 56)

server ([section 3.1.4](#) 26, [section 3.3.4](#) 52)

Server

abstract data model ([section 3.1.1](#) 21, [section 3.3.1](#) 50)

initialization ([section 3.1.3](#) 25, [section 3.3.3](#) 51)

local events ([section 3.1.6](#) 49, [section 3.3.6](#) 55)

message processing ([section 3.1.4](#) 26, [section 3.3.4](#) 52)

[Microsoft Office SharePoint Server interface](#) 21

[overview - Microsoft Office SharePoint Server](#) 21

[overview - Windows SharePoint Services](#) 50

[proc_MSS_FetchSDs method](#) ([section 3.1.4.1](#) 28, [section 3.3.4.1](#) 53)

[proc_MSS_GetConfigurationProperty method](#) ([section 3.1.4.2](#) 28, [section 3.3.4.2](#) 53)

[proc_MSS_GetKeywordInformation method](#) 29

[proc_MSS_GetManagedProperties method](#) ([section 3.1.4.3](#) 28, [section 3.3.4.3](#) 53)

[proc_MSS_GetManagedPropertyAliases method](#) ([section 3.1.4.4](#) 29, [section 3.3.4.4](#) 53)

[proc_MSS_GetMultipleResults method](#) ([section 3.1.4.6](#) 30, [section 3.3.4.5](#) 53)
[proc_MSS_GetSchemaHighLevelInfo method](#) ([section 3.1.4.7](#) 33, [section 3.3.4.6](#) 54)
[proc_MSS_GetSchemaParameters method](#) ([section 3.1.4.8](#) 33, [section 3.3.4.7](#) 54)
[proc_MSS_GetVectorI4ConfigurationProperty method](#) ([section 3.1.4.9](#) 33, [section 3.3.4.8](#) 55)
[proc_MSS_QLog_AllocateClickIds method](#) 34
[proc_MSS_QLog_DeleteOld method](#) 34
[proc_MSS_QLog_InsertQueryInfo method](#) 35
[proc_MSS_QLog_LowClickThrough method](#) 35
[proc_MSS_QLog_QueriesByDay method](#) 36
[proc_MSS_QLog_QueriesByMonth method](#) 37
[proc_MSS_QLog_QueriesByScope method](#) 38
[proc_MSS_QLog_TopBestBets method](#) 39
[proc_MSS_QLog_TopDestinations method](#) 40
[proc_MSS_QLog_TopQueries method](#) 41
[proc_MSS_QLog_TopSiteCollections method](#) 42
[proc_MSS_QLog_ZeroBestBets method](#) 43
[proc_MSS_QLog_ZeroResults method](#) 43
[proc_MSS_RetrieveDuplicateHashes method](#) ([section 3.1.4.23](#) 45, [section 3.3.4.9](#) 55)
[proc_MSS_RetrieveDuplicateHashesEx method](#) ([section 3.1.4.24](#) 45, [section 3.3.4.10](#) 55)
[proc_MSS_ScopeCacheGetChangedConsumers method](#) 46
[proc_MSS_ScopeCacheGetChanges method](#) 47
[sequencing rules](#) ([section 3.1.4](#) 26, [section 3.3.4](#) 52)
[timer events](#) ([section 3.1.5](#) 48, [section 3.3.5](#) 55)
[timers](#) ([section 3.1.2](#) 25, [section 3.3.2](#) 51)
[Windows SharePoint Services interface](#) 50
Simple data types
Advanced Search Flag ([section 2.2.1.4](#) 14, [section 2.2.1.4](#) 14)
Continued Flag ([section 2.2.1.5](#) 14, [section 2.2.1.5](#) 14)
DidYouMean Flag ([section 2.2.1.6](#) 15, [section 2.2.1.6](#) 15)
IsSSPLLevel Flag ([section 2.2.1.8](#) 15, [section 2.2.1.8](#) 15)
[overview](#) 14
ResultView Flag ([section 2.2.1.7](#) 15, [section 2.2.1.7](#) 15)
Scope ([section 2.2.1.3](#) 14, [section 2.2.1.3](#) 14)
Site Collection Identifier ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
Site Identifier ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
Site Collection Identifier simple type ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
Site Identifier simple type ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
[Standards assignments](#) 13
Structures
[binary](#) 15
[table and view](#) 15
[XML](#) 20
Summary Query Log Set

[abstract data model](#) 25

T

Table structures
[#Temp](#) 19
[MSSDocProps](#) 16
[MSSDocSdids](#) 16
[MSSDuplicateHashes](#) 17
[MSSQLogLocation](#) 19
[MSSQLogNonClickedUnprocessed](#) 19
[MSSQLogUnprocessed](#) 17
[Table structures - overview](#) 15
Timer events
client ([section 3.2.5](#) 50, [section 3.4.5](#) 56)
server ([section 3.1.5](#) 48, [section 3.3.5](#) 55)
Timers
client ([section 3.2.2](#) 50, [section 3.4.2](#) 56)
server ([section 3.1.2](#) 25, [section 3.3.2](#) 51)
[Tracking changes](#) 64
[Transport](#) 14

V

[Vendor-extensible fields](#) 13
[Versioning](#) 13
View structures
[#Temp](#) 19
[MSSDocProps](#) 16
[MSSDocSdids](#) 16
[MSSDuplicateHashes](#) 17
[MSSQLogLocation](#) 19
[MSSQLogNonClickedUnprocessed](#) 19
[MSSQLogUnprocessed](#) 17
[View structures - overview](#) 15

W

[Windows SharePoint Services interface - client](#) 55
[Windows SharePoint Services interface - server](#) 50

X

[XML structures](#) 20