[MS-FSQR]:
Query and Result Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting 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
11/06/2009	0.1	Major	Initial Availability
02/19/2010	1.0	Minor	Updated the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	Minor	Clarified the meaning of the technical content.

Table of Contents

1   Introduction

This document specifies the Query and Result Protocol. A protocol client uses this protocol to submit search queries to a protocol server for indexed content and to receive search results from that protocol server.

1.1   Glossary

The following terms are defined in [MS-GLOS]:

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

The following terms are defined in [MS-OFCGLOS]:

deep refinement
document summary
document vector
dynamic rank
endpoint
FAST Search Authorization (FSA)
field collapsing
freshness boost
index partition
index schema
managed property
query matching node
query refinement
query transform
rank
rank profile
refinement bin
search service application
SOAP (Simple Object Access Protocol)
stemming
summary class
WSDL operation

The following terms are specific to this document:

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

1.2   References

1.2.1   Normative References

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

[ISO-639] International Organization for Standardization, "Codes for the Representation of Names of Languages", ISO 639, http://www.loc.gov/standards/iso639-2/

[MC-NMF] Microsoft Corporation, ".NET Message Framing Protocol Specification", October 2007.

[MS-FSCF] Microsoft Corporation, "Content Feeding Protocol Specification", November 2009.

[MS-FSDQE] Microsoft Corporation, "Distributed Query Execution Protocol Specification", November 2009.

[MS-FSFQL] Microsoft Corporation, "Fast Query Language Structure", November 2009.

[MS-FSSAS] Microsoft Corporation, "Search Authorization Synchronization Protocol Specification", February 2010.

[MS-FSSCFG] Microsoft Corporation, "Search Configuration File Format Specification", November 2009.

[MS-SEARCH] Microsoft Corporation, "Search Protocol Specification", June 2008.

[MS-SPSTWS] Microsoft Corporation, "SharePoint Security Token Service Web Service Protocol Specification", July 2009.

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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part1-20030624

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, http://www.w3.org/TR/REC-xml-names/

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2   Informative References

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

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

1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to submit search requests to a protocol server on behalf of a user.

A search request is comprised of a query expression and query parameters. Query parameters specify general aspects of the search, including:

- The natural language of the query, such as English, Norwegian, or Japanese.

- Whether the specified conditions are all necessary versus whether any matching condition is sufficient for an item to be included in the search result.

- The effect of the age of an item on its **dynamic rank**.

- How to adjust the dynamic rank of an item based on its similarity to a specified **document vector**.

- Whether to include or exclude an item based on its similarity to a specified document vector.

- How to sort the items in the search result.

- Whether to consider alternative forms of words when matching them to items.

- Whether to correct the spelling of words in the search expression.

- Whether to apply **query refinement**.

- Whether to perform **field collapsing**.

- The dynamic criteria for identifying and removing redundant results.

The protocol server returns a search result to the protocol client that includes the items meeting the search criteria and that excludes any items that the user is unauthorized to view.

1.4 Relationship to Other Protocols

The Query and Result Protocol uses **SOAP** over the .NET Message Framing Protocol (as described in [MC-NMF]), as shown in the following layering diagram:
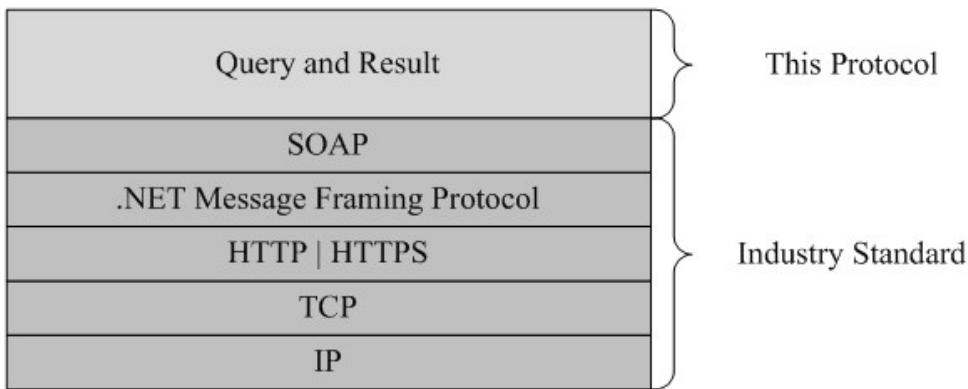
Figure 1: This protocol in relation to other protocols

Release: Sunday, December 19, 2010

Queries that use this protocol are expressed in either SharePoint Search Keyword Syntax, as described in [MS-SEARCH] section 2.2.9, or FAST Query Language (FQL), as described in [MS-FSFQL].

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a **well-known endpoint**.

1.6 Applicability Statement

This protocol is designed to enable a client application to submit a search request to and receive a search result from a protocol server. This protocol is applicable when a client application needs to search for items that are already known to a protocol server.

1.7 Versioning and Capability Negotiation

For versioning issues and capability negotiation, this protocol uses SOAP as specified in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

All protocol messages MUST be transported by means of HTTP or HTTPS.

Protocol messages MUST be formatted according to either SOAP 1.1, as specified in [SOAP1.1] section 4, or SOAP 1.2, as specified in [SOAP1.2/1] section 5.

Communications MUST use binary encoding as specified in [MC-NMF] section 2.2.3.4.1.

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses XML Schema as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language as specified in [WSDL].

2.2.1 Namespaces

This specification defines and references various XML namespaces using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

2.2.2 Messages

None.

2.2.3 Elements

This specification does not define any common XML Schema element definitions.

2.2.4 Complex Types

This specification does not define any common XML Schema complex type definitions.

2.2.5 Simple Types

This specification does not define any common XML Schema simple type definitions.

2.2.6 Attributes

This specification does not define any common XML Schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML Schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML Schema attribute group definitions.

3   Protocol Details

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

3.1   Query and Result Protocol Server Details

The protocol server MUST respond to **ProcessRequest** operations from protocol clients with a search result or an error.

Search results MUST be formatted as specified in section 3.1.4.1.2.2.1. To control how the request will be processed and to generate the query request, as specified in [MS-FSDQE] section 2.2.6, the protocol server MUST evaluate the parameters as specified in section 3.1.4.1.2.1.1.

The **FAST Search Authorization (FSA)** worker component of the protocol server MUST use the authorization information that is synchronized by the FSA manager component as specified in [MS-FSSAS] to ensure that the search result includes only the items that the specified user is authorized to view, according to the **docacl** attribute as specified in [MS-FSCF] section 2.2.38.

The user identity MUST be obtained by using claims, as specified in [MS-SPSTWS].

3.1.1   Abstract Data Model

None.

3.1.2   Timers

None.

3.1.3   Initialization

The protocol server MUST create a listening endpoint that provides the **qrsproxyClaims** service as specified in section 6.

3.1.4   Message Processing Events and Sequencing Rules

The single **WSDL operation** for the protocol server is the **ProcessRequest** operation, which first uses the specified query to perform a search and then returns a result string.

3.1.4.1   ProcessRequest

The **ProcessRequest** operation submits a search query and retrieves the corresponding search result.

```
<wsdl:operation name="ProcessRequest">
  <wsdl:input
wsaw:Action="http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequest
" message="tns:IProxyRemote_ProcessRequest_InputMessage" />
  <wsdl:output
wsaw:Action="http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequest
Response" message="tns:IProxyRemote_ProcessRequest_OutputMessage" />
</wsdl:operation>
```

The protocol client MUST send an **IProxyRemote_ProcessRequest_InputMessage** message, and the protocol server MUST respond with an **IProxyRemote_ProcessRequest_OutputMessage** message.

The call to the **ProcessRequest** operation MUST use one the following **endpoints (3)**:

- http://###SERVER_ADDRESS###/qrsproxyClaims

- https://###SERVER_ADDRESS###/qrsproxyClaims

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.1.1.1 IProxyRemote_ProcessRequest_InputMessage

A protocol client sends the **IProxyRemote_ProcessRequest_InputMessage** message to submit a search query.

The SOAP action value of this message MUST be:

```
http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequest
```

The SOAP body contains a **ProcessRequest** element.

3.1.4.1.1.2 IProxyRemote_ProcessRequest_OutputMessage

The protocol client sends the **IProxyRemote_ProcessRequest_OutputMessage** message to request a search result.

The SOAP action value of the message MUST be:

```
http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequestResponse
```

The SOAP body contains a **ProcessRequestResponse** element.

3.1.4.1.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.1.2.1 ProcessRequest

The **ProcessRequest** element contains the search query.

```
<xs:element name="ProcessRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="request" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

request: A valid search query, as specified in section 3.1.4.1.2.1.1.

3.1.4.1.2.1.1 Search Query

A search query MUST begin with the string "http://q/cgi-bin/search?" and have parameters appended to that string.

The list of parameters MUST form a valid query component, as specified in [RFC2396] section 3.4. Within the parameter values, any characters that are reserved or excluded from the printable US-ASCII character set MUST be escaped as specified in [RFC2396] section 2.4. Parameters MUST be delimited by an ampersand character ("&") that is not escaped.

The parameter names MUST be lowercase.

If a parameter of type **Flag** is specified without a value, it MUST be assumed that the parameter has a value of 1. If a parameter is of any other type, the parameter name MUST be immediately followed by an equal sign ("=") and then the parameter value.

3.1.4.1.2.1.1.1 Query Parameter

The query parameter specifies the search query to evaluate.

Unless overridden by the type parameter, as specified in section 3.1.4.1.2.1.1.2, the value of the query parameter MUST be a valid FQL expression, as specified in [MS-FSFQL].

3.1.4.1.2.1.1.2 Query Type Parameter

The type parameter overrides the default format for the query parameter.

The value of the *type* parameter MUST be one that is listed in the following table.

Value	Description
"kwany"	The query parameter MUST be expressed in SharePoint Search Keyword Syntax, as specified in [MS-SEARCH] section 2.2.9. For an item to be included in the search result, at least one of the keywords from the query MUST be matched.
"kwall"	The query parameter MUST be expressed in SharePoint Search Keyword Syntax, as specified in [MS-SEARCH] section 2.2.9. For an item to be included in the search result, all the keywords of the query MUST be matched.

If the type parameter is omitted, the value of the query parameter MUST be a valid FQL expression, as specified in [MS-FSFQL].

3.1.4.1.2.1.1.3 Filter Parameter

The qtf_parsekw:filter parameter specifies additional search criteria that MUST be evaluated in conjunction with the query parameter. Both the filter and the query MUST match all the items that are included in the search result. The value of the qtf_parsekw:filter parameter MUST be a valid FQL expression, and it MUST be evaluated in the same way as the FQL **filter** operator, as specified in [MS-FSFQL].

The qtf_parsekw:filter parameter MUST be ignored unless the type parameter is set to "kwall".

3.1.4.1.2.1.1.4   Freshness Boost Parameter

The *qtf_freshnessboost:datetime* parameter specifies a point in time to be used as a basis for the **freshness boost** calculation. The value MUST be a valid **datetime-value** value, as specified by the following **Augmented Backus-Naur Form (ABNF)** rules:

```
datetime-value = year "-" month "-" day
    ["T" hour ":" minute ":" second ["Z"]]
year = 4*DIGIT              ; 0000-infinity - four-digit (or longer) year
month = ("0" DIGIT)         ; 00-09
    / ("1" %x30-32)         ; 10-12 (00-12) - two-digit month
day = (%x30-32 DIGIT)       ; 00-29
    / ("3" %x30-31)         ; 30-31 (00-31) - two-digit day
hour = (%x30-31 DIGIT)      ; 00-19
    / ("2" %x30-33)         ; 20-23 (00-23) - two-digit hour
minute = (%x30-35 DIGIT)    ; 00-59        - two-digit minute
second = (%x30-35 DIGIT)    ; 00-59        - two-digit second
```

3.1.4.1.2.1.1.5   Similarity Parameters

The similarity parameters specify whether to find, exclude, or refine the dynamic rank of similar items.

Given the document vector of an item from a previous search result, the similarity parameters specify the impact of similarity on the results—that is, whether to find similar items, exclude similar items, or refine the dynamic rank of similar items. The similarity parameters are described in the following table.

Parameter name	Type	Description
rpf_sortsimilar:enabled	Flag	Specifies whether to use document vectors to refine dynamic ranks. A value of 1 means yes, and a value of 0 means no.
similartype	String	Specifies the type of similarity. This parameter is applicable only if rpf_sortsimilar:enabled has a value of 1. The valid values for this parameter are described later in this section.
similarto	String	Specifies one or more document vectors, each in the form **[***term***,***weight***]**, where *term* MUST be a single word or phrase, and *weight* MUST be a real number from 0 through 1 that indicates the degree of dynamic rank. To find the items that are similar to an item from a previous search result, use the value of the item's document vector as the value of the *similarto* parameter. The similarto parameter is applicable only if rpf_sortsimilar:enabled has a value of 1.

The value of the *similartype* parameter MUST be one that is listed in the following table.

Value	Description
"find"	Includes items that match the query, the *similarto* parameter, or both. That is, the *similarto* parameter acts as an **OR** condition to the query.

Value	Description
"refine"	Includes items that match both the query and the *similarto* parameter. That is, the *similarto* parameter acts as an **AND** condition to the query.
"exclude"	Includes items that match the query but not the *similarto* parameter. That is, the *similarto* parameter acts as an **AND NOT** condition to the query.

3.1.4.1.2.1.1.6 Sorting Parameters

The sorting parameters specify the sorting criteria for the search result. By default, search results are sorted by **rank** in ascending order. The following table describes the sorting parameters.

Parameter name	Type	Description
sortdirection	**String**	Specifies the default sort direction. The valid values for this parameter are described later in this section.
sortby	**String**	Specifies the criteria for sorting, as specified in section 3.1.4.1.2.1.1.6.1.
random	**Integer**	Specifies that the 4000 highest-ranked items MUST be returned in pseudo-random order, with the provided integer value used as the random seed. The order that results from a given seed is implementation-dependent, but unless the search index is updated, the same random seed MUST result in the same item order.

The value of the *sortdirection* parameter MUST be one that is listed in the following table.

Value	Description
"ascending"	Sorts in ascending order (smallest first, largest last).
"descending"	Sorts in descending order (largest first, smallest last). This value acts as the default.

3.1.4.1.2.1.1.6.1 sortby Parameter

The *sortby* parameter specifies the sorting criteria and MUST conform to the following ABNF rules:

```
sortby = sort-level *(" " sort-level)

sort-level = [sort-direction] (managed-property / rank-profile / rank
    / random / formula)
managed-property = 1*(ALPHA / DIGIT)
rank-profile = 1*(ALPHA / DIGIT)
rank = "[rank]"
random = "[random:seed=" random-seed
    [":hashfield=" hashfield] [":addtorankmax=" addtorankmax] "]"
formula = "[formula:" expression "]"

sort-direction = ascending / descending
ascending = "+"
descending = "-"
```

```
random-seed = unsigned-integer
hashfield = managed-property
addtorankmax = integer
integer = ["-" / "+"] 1*DIGIT
unsigned-integer = 1*DIGIT

expression = (function / managed-property / rank
    / parenthetic-expression / number)
    *(operator expression)
parenthetic-expression = "(" expression ")"
function = unary-function / pow / atan2 / bin
unary-function = ("sqrt" / "exp" / "log" / "abs" / "ceil" / "floor"
    / "round" / "sin" / "cos" / "tan" / "asin" / "acos" / "atan")
    "(" expression ")"
pow = "pow(" expression "," expression ")"
atan2 = "atan2(" expression "," expression ")"
bin = "bucket(" expression *("," bin-limit) ")"
bin-limit = number
number = (1*DIGIT ["." 1*DIGIT]) / ("." 1*DIGIT)
operator = "*" / "/" / "+" / "-"
```

The sortby parameter specifies one or more sort levels, delimited by spaces, in decreasing order of significance. In the preceding ABNF rules, a sort level is specified by the **sort-level** rule.

To override the default sort direction of "descending", the sort direction MUST be specified either by using the **sort-direction** element as the prefix in the **sort-level** rule or by using the sortdirection parameter.

Each **sort-level** element MUST specify one of the elements that are described in the following table.

Element	Description
managed-property	The items in the search result MUST be sorted by the specified **managed property**, and the **SortableType** member of that managed property MUST be set to "Enabled" as specified in [MS-FSSCFG] section 1.3.2.1.
rank-profile	The items in the search result MUST be sorted by the specified **rank profile**.
rank	The items in the search result MUST be sorted by dynamic rank.
random	The items in the search result MUST exist in pseudo-random order.
formula	The items in the search result MUST be sorted by the numeric result of the specified **expression** element.

The **random** element specifies that all the items that match the query MUST be sorted in random order, rather than just the highest-ranking 4000 items that the random parameter specifies. The options for the **random** element are described in the following table.

Element	Description
random-seed	The integer seed to use when generating the pseudo-random order for the search result.
hashfield	The name of a managed property that contains random or unique values. This name MUST guarantee the preservation of the sort order even after a search index update. The specified managed property MUST be defined as a 64-bit integer in the **index**

Element	Description
	schema.
addtorankmax	The maximum pseudo-random integer value that MUST be added to the rank of each item to further increase randomness for the purpose of sorting. This value MUST NOT impact the dynamic ranks that are returned in the search result.

If the **formula** element is specified, the **expression** element MUST be evaluated for matching items. The evaluation MUST occur left-to-right and use standard mathematical-operator precedence. That is, functions and parenthetical groups MUST be evaluated first, multiplication and division operations MUST be performed next, and addition and subtraction operations MUST be performed last.

The **expression** element MUST NOT contain spaces.

The **expression** element supports the functions that are listed in the following table.

Function	Description
sqrt(n**)**	The square root of n.
exp(n**)**	The exponential function that is equivalent to **pow(2.71828182846,**n**)**.
log(n**)**	The natural logarithm of n.
abs(n**)**	The absolute value of n.
ceil(n**)**	The ceiling of n. That is, if n is not a whole number, round up to the next whole number. If n is a whole number, use n.
floor(n**)**	The floor of n. That is, if n is not a whole number, round down to the next whole number. If n is a whole number, use n.
round(n**)**	The rounding of n to the nearest whole number.
sin(n**)**	The sine of n radians.
cos(n**)**	The cosine of n radians.
tan(n**)**	The tangent of n radians.
asin(n**)**	The arcsine, in radians, of n.
acos(n**)**	The arccosine, in radians, of n.
atan(n**)**	The arctangent, in radians, of n.
pow(x,y**)**	The value of x raised to the power of y.
atan2(y,x**)**	A two-argument arctangent—the angle in radians between the positive x axis and the specified Cartesian coordinate (x,y).
bucket(b,$n1$,…**)**	An arbitrary number of **refinement bins** for the **expression** element b. Values that follow b (that is, $n1$, $n2$, $n3$, and so forth) are numbers that specify refinement bin names and limits.
	The lowest bin value ($n1$ if bins are specified in ascending order) MUST contain all the items for which b evaluates to a number that is less than $n1$. Subsequent refinement bins follow the same rule but MUST exclude the items that were included in previous bins. Values greater than the highest specified bin limit MUST be included in the

Function	Description
	highest bin.

3.1.4.1.2.1.1.7   Stemming Parameter

The *qtf_lemmatize* parameter specifies whether to search for alternative forms of words. This parameter is of type **Flag** and specifies whether **stemming** is enabled for the query.

3.1.4.1.2.1.1.8   Spell-Checking Parameter

The *spell* parameter specifies whether to spell-check and correct the spelling in the query. The following table lists the valid values for the *spell* parameter.

Value	Description
"on"	Enables spell-checking and correction for the query
"suggest"	Specifies that the result will contain the corrected query but that the query will be executed without correction
"off"	Disables spell-checking and correction

To correct the spelling of a phrase, specify the named parameter *mode*, as specified in [MS-FSFQL] section 2.1.17.2, as follows:

```
mode="phrase"
```

3.1.4.1.2.1.1.9   Automatic Resubmission Parameter

The *resubmitflags* parameter contains a bitmask that specifies the criteria for automatically resubmitting a query that yielded no results. The following table lists the valid values for the *resubmitflags* parameter.

Value	Description
1 (0x00000001)	Do not resubmit.
64 (0x00000040)	Resubmit with stemming enabled.
7984 (0x00001f30)	Resubmit with spell-checking enabled.
8192 (0x00002000)	Resubmit with spell-checking in **suggest** mode. Spell-checking MUST be enabled for this value to have an effect. Therefore, this value MUST be combined with the previous value (7984) by using a bitwise **AND** operator, which results in 16176.
134217728 (0x08000000)	Resubmit with synonyms added.

Any bitwise combination of valid *resubmitflags* values is also valid. However, if the 1 bit is set, all the other bits will be ignored and the query will not be resubmitted.

3.1.4.1.2.1.1.10   Query Refinement Parameters

The query refinement parameters specify criteria for navigation by using managed properties from the search result. The following table describes all the query refinement parameters.

Parameter name	Type	Description
rpf_navigation:enabled	**Flag**	Whether query refinement is enabled for the query.
rpf_navigation:navigators	**String**	A comma-delimited list of refiners to be included with the search result. A detailed description of the format for this parameter appears later in this section. Valid refiners are defined by the refiner index schema, as specified in [MS-FSSCFG] section 2.5.
rpf_navigation:hits	**Integer**	The number of items to use when creating refiners.

The value of rpf_navigation:navigators MUST be formatted according to the following ABNF rules:

```
rpf-navigation-navigators-value = refiner *("," refiner)

refiner = refiner-name ["(" [parameter *("," parameter)] ")"]
refiner-name = index-schema-element-name
parameter = sort / filter / deephits / cutoff

sort = "sort=" property "/" direction
property = "frequency" / "name" / "number"
direction = "descending" / "ascending"

filter = "filter=" max-bins "/" min-frequency "/"
    bin-name-prefix
max-bins = unsigned-integer-value
min-frequency = unsigned-integer-value
bin-name-prefix = "*" / (1*bin-name-char)
bin-name-char = ALPHA / DIGIT / WSP

deephits = "deephits=" max-hits
max-hits = unsigned-integer-value
cutoff = "cutoff=" min-frequency "/" min-bins "/" max-bins
min-bins = unsigned-integer-value
index-schema-element-name = 1*(ALPHA / DIGIT)
unsigned-integer-value = 1*DIGIT
```

The options for the **filter** element are described in the following table.

Element	Description
max-bins	The refiner MUST NOT return more refinement bins than the value of **max-bins**.
min-frequency	The refiner MUST NOT return any refinement bin that has a frequency value less than that of **min-frequency**.
bin-name-prefix	The refiner MUST NOT return any refinement bin unless its name begins with the value of **bin-name-prefix**.

The options for the **sort** element are described in the following table.

Value	Description
"frequency"	The refiner MUST sort its refinement bins by number of items.
"name"	The refiner MUST sort its refinement bins by name.
"number"	The refiner MUST sort numerically by refinement bin name. For example, the name "05" MUST be considered greater than the name "4".

The **deephits** element specifies that **deep refinement** MUST NOT use more hits than the value of the **max-hits** element specifies.

The **cutoff** element applies only to deep refinement. The options for the **cutoff** element are described in the following table.

Element	Description
min-frequency	The refiner MUST NOT return refinement bins with frequency values less than that of **min-frequency**.
min-bins	The refiner MUST NOT return fewer than refinement bins than the value of **min-bins**.
max-bins	The refiner MUST NOT return more refinement bins than the value of **max-bins**.

3.1.4.1.2.1.1.11   Field Collapsing Parameters

The field collapsing parameters specify which numeric managed properties will be used for field collapsing. The following table describes the field collapsing parameters.

Parameter name	Type	Description
collapseon	**String**	The managed property name to collapse. The managed property name MUST be prefixed with "batv".
collapsenum	**Integer**	The number of items to include from the collapsing process. The default value is 1.

3.1.4.1.2.1.1.12   Dynamic Duplicate Removal Parameters

The dynamic duplicate removal parameters specify criteria for the identification and removal of duplicates in the search result. The following table describes the dynamic duplicate removal parameters.

Parameter name	Type	Description
rff_ddr:enabled	**Flag**	Whether to enable or disable dynamic duplicate removal.
rff_ddr:slot1	**String**	The primary slot for specifying managed properties that the query processing component uses for dynamic duplicate removal.
rff_ddr:slot2	**String**	The secondary slot for dynamic duplicate removal criteria.

Each managed property that is used for dynamic duplicate removal MUST be defined in the index schema as of type **Text**, as specified in [MS-FSSCFG] section 2.1.1.

The query processing component MUST remove duplicates based independently on the criteria that is specified in either *rff_ddr:slot1* or *rff_ddr:slot2*.

The values of *rff_ddr:slot1* and *rff_ddr:slot2* MUST be formatted according to the following ABNF rules:

```
ddr-slot = ddr-condition *("," ddr-condition)
ddr-condition = preferred-managed-property
    ["/" fallback-managed-property]

preferred-managed-property = "bsum" managed-property-name
fallback-managed-property = "bsum" managed-property-name

managed-property-name = 1*(ALPHA / DIGIT)
```

The query processing component MUST remove an item as a duplicate if another item in the search result contains equal values for each managed property that it evaluates.

For each item, the query processing component MUST evaluate the managed property that is specified by the **preferred-managed-property** element, unless the preferred managed property is empty for that item and the **ddr-condition** element specifies a **fallback-managed-property** element.

If the preferred managed property is empty and a fallback managed property is specified, the query processing component MUST evaluate the fallback managed property.

3.1.4.1.2.1.1.13   Query Language Parameter

The language parameter specifies the target language for the query, as specified in [ISO-639]. The default value is "en".

3.1.4.1.2.1.1.14   White Space Preservation Parameter

The *tvmarg9* parameter specifies whether white space characters are to be preserved in the query result. A value of 1 enables white space preservation, and a value of 0 disables such preservation.

This parameter is applicable only if the **document summary** preserves formatting characters.

3.1.4.1.2.1.1.15   Result Count Limit Parameter

The hits parameter specifies the maximum number of items to be returned in the search result. The value of the hits parameter MUST be a positive integer. The default value is 10.

3.1.4.1.2.1.1.16   Result Offset Parameter

The *offset* parameter specifies how many items to skip before beginning the search result. The value of the *offset* parameter MUST be an integer from 0 through 4020. The default value is 0.

Used in combination with the *hits* parameter, the *offset* parameter permits result retrieval a page at a time.

3.1.4.1.2.2   ProcessRequestResponse

The **ProcessRequestResponse** element contains the search result.

```
<xs:element name="ProcessRequestResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProcessRequestResult" type="tns:StreamBody" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ProcessRequestResult: An element that MUST be a valid search result, as specified in section 3.1.4.1.2.2.1.

3.1.4.1.2.2.1 Search Result

The search result contains items that matched the search query, an indication that no items matched the search query, or an indication that an error occurred.

3.1.4.1.2.2.1.1 Success

The result of a search that matches one or more items MUST be formatted according to the following ABNF rules:

```
search-result =
    sr-header
    [qt-section]
    [refinement-section]
    item-section
    paging-section

; Search result header

sr-header = srh-segment-name srh-mta-sep section-separator

srh-segment-name = "#SEG NAM webcluster" CRLF
srh-mta-sep = "#MTA SEP" vwsp list-entry-separator CRLF
list-entry-separator = anychar

; Query transform section

qt-section = 1*qt

qt = qt-header qt-body section-separator

qt-header = qtp-name qtp-action
qtp-name = qtp-prefix "NAM" vwsp string-value CRLF
qtp-action = qtp-prefix "ACT" vwsp string-value CRLF

qt-body = *qt-property
qt-property = qtp-string / qtp-msg
qtp-string = qtp-prefix qt-string-prop-name [vwsp string-value] CRLF
qt-string-prop-name = "QRY" / "CUS"

; When present, MSG always has an accompanying MID.
qtp-msg = qtp-prefix "MSG" [vwsp string-value] CRLF
    qtp-prefix "MID" vwsp unsigned-integer-value CRLF

qtp-prefix = "#QTF" vwsp
```

```
    ; Refinement section

    ; The exact number of <refiner> occurrences in <refinement-section>
    ; is specified within <refiner-header> by the <unsigned-integer-value>
    ; of <refiner-count>.
    refinement-section = refinement-section-header *refiner

    refinement-section-header = refiner-count section-separator
    refiner-count = rprop-prefix "ENT" vwsp unsigned-integer-value CRLF

    refiner = refiner-body section-separator

    refiner-body = 1*rprop

    ; Refiner properties
    rprop-prefix = "#NAV" vwsp

    rprop = rprop-prefix (rprop-sep-list / rprop-string / rprop-int
        / rprop-float / rprop-refiner-type / rprop-min-max-mean) CRLF

    rprop-sep-list = sep-list-rprop-name [vwsp sep-list-value]
    rprop-string = string-rprop-name [vwsp string-value]
    rprop-int = int-rprop-name vwsp integer-value
    rprop-float = float-rprop-name vwsp float-value
    rprop-refiner-type = "TYPE" vwsp
        ("Datetime" / "Discretenumeric" / "Double" / "Integer" / "String")
    rprop-min-max-mean = min-max-mean-rprop-name vwsp
        (datetime-value / float-value)

    sep-list-rprop-name = "NAMES" / "MODS" / "CNTS"
    string-rprop-name = "NAME" / "DNAM" / "MODI" / "UNIT"
    int-rprop-name = "UCNT" / "HCNT" / "SCNT"
    float-rprop-name = "ETPY" / "SCOR" / "RATI" / "SUM"
    min-max-mean-rprop-name = "MIN" / "MAX" / "MEAN"

    ; Item section

    item-section = item-section-header ((1*item) / no-items)

    item-section-header = ish-prop item-section-header-end
    item-section-header-end = "###/" CRLF

    ish-prop = ish-first ish-last ish-hits ish-count ish-time ish-max-rank

    ; Item section header properties
    ish-first = "#FIR" vwsp unsigned-integer-value CRLF
    ish-last = "#LAS" vwsp unsigned-integer-value CRLF
    ish-hits = "#HTS" vwsp unsigned-integer-value CRLF
    ish-count = "#CNT" vwsp unsigned-integer-value CRLF
    ish-time = "#TIM" vwsp float-value CRLF
    ish-max-rank = "#MAR" vwsp unsigned-integer-value CRLF

    item = item-header *item-managed-property item-end
    item-header = item-start *(item-property / item-managed-property)
    item-property = "#" ("rank" / "fcoid" / "fcocount" / "morehits" )
        vwsp unsigned-integer-value CRLF

    item-start = "####" vwsp unsigned-integer-value CRLF
    item-end = "###/" CRLF
```

```
item-managed-property = "#" managed-property-name [vwsp string-value]
    CRLF
managed-property-name = 1*(ALPHA / DIGIT)

no-items = total-match-count query-time
total-match-count = "#C" vwsp unsigned-integer-value CRLF
query-time = "#T" vwsp float-value CRLF

; Paging section

paging-section = *CRLF *1(prev-page) *1(next-page) *CRLF
next-page = "#N" vwsp page-spec CRLF
prev-page = "#P" vwsp page-spec CRLF
page-spec = page-startindex vwsp page-endindex vwsp page-query
page-startindex = unsigned-integer-value
page-endindex = unsigned-integer-value
page-query = string-value

; Data types

string-value = *(WSP/VCHAR/ichar)
integer-value = ["-" / "+"] 1*DIGIT
unsigned-integer-value = 1*DIGIT
float-value = ["-" / "+"] 1*DIGIT ["." 1*DIGIT]

datetime-value = year "-" month "-" day
    ["T" hour ":" minute ":" second ["." sub-second] ["Z"]]
year = 4*DIGIT              ; four-digit or longer year (0000-infinity)
month = ("0" DIGIT)        ; two-digit month (00-09)
    / ("1" %x30-32)        ; two-digit month (10-12)
day = (%x30-32 DIGIT)      ; two-digit day (00-29)
    / ("3" %x30-31)        ; two-digit day (30-31)
hour = (%x30-31 DIGIT)     ; two-digit hour (00-19)
    / ("2" %x30-33)        ; two-digit hour (20-23)
minute = (%x30-35 DIGIT)   ; two-digit minute (00-59)
second = (%x30-35 DIGIT)   ; two-digit second (00-59)
sub-second = 1*7(DIGIT)    ; subseconds, up to seven digits

; <sep-list-value> is a list of values delimited by the character
; specified in <srh-mta-sep>.
sep-list-value = *anychar

; General syntax elements

section-separator = "####" CRLF

vwsp = 1*WSP  ; variable-length white space

; <ichar> can be any international character (not US-ASCII).
ichar = %x7f-ffffffff

; <anychar> can be any character except CR or LF.
anychar = %x00-09 / %x0b-0c / %x0e-ffffffff
```

Although ABNF, as specified in [RFC5234], does not explicitly support any encoding other than US-ASCII, the **ichar** and **anychar** elements support wide character values that are encoded as **UTF-8**.

Release: Sunday, December 19, 2010

3.1.4.1.2.2.1.1.1   Search Result Header

A search result header contains two lines: one for the **srh-segment-name** element and another for the **srh-mta-sep** element.

The **srh-segment-name** element MUST contain "#SEG NAM webcluster". The **srh-mta-sep** element specifies the **list-entry-separator** character that is used as the delimiter in values of type **sep-list-value**.

3.1.4.1.2.2.1.1.2   Query Transform Section

Query transform properties—that is, **qt-property** elements in the **qt-section** element—are informational and ignored by the protocol client. These properties are described in the following table.

Property	Description
"NAM"	The query transform name.
"ACT"	Text that describes the query transformation action.
"QRY"	The query as represented internally after some processing. This property contains implementation-specific data and MUST be discarded.
"CUS"	A custom property—the meaning varies based on the query transform.
"MSG"	A text message that indicates status.
"MID"	The numeric identifier of the message that is represented in the "MSG" property.

The valid values for the "NAM" property are as follows:

- "Original query"

- "FastQT_Keyword"

- "FastQT_Lemmatizer"

- "Final query"

The following table lists the valid values for the "ACT" property, depending on the value of the "NAM" property.

Value of "NAM" property	Value of "ACT" property
"Original query"	"NOP"
"FastQT_Keyword"	"Suggested new query"
"FastQT_Lemmatizer"	"nop"
"Final Query"	"NOP"

The following table lists the valid values for the "CUS" property, depending on the value of the "NAM" property.

Value of "NAM" property	Value of "CUS" property
"Original query"	(none)
"FastQT_Keyword"	(none)
"FastQT_Lemmatizer"	"No change to query"
"FastQT_Lemmatizer"	"No lemmatizer for query language"
"FASTQT_Lemmatizer"	"Modified query stack"
"FASTQT_Lemmatizer"	"No change to query stack"
"Final Query"	"FQL"

The following table lists the valid "MID" and "MSG" value pairs, depending on the value of the "NAM" property.

Value of "NAM" property	"MID" value	"MSG" value
"Original query"	(none)	(none)
"FastQT_Keyword"	1	"Keyword processing"
"FastQT_Lemmatizer"	1	"Lemmatized term (query reduction)"
"FastQT_Lemmatizer"	2	"Lemmatized term (query expansion)"
"FastQT_Lemmatizer"	4	"Term rerouted (lemmatization by document expansion)"
"FastQT_Lemmatizer"	8	"No query transformation"
"FastQT_Lemmatizer"	16	"Lemmatization turned off for current query"
"FastQT_Lemmatizer"	32	"Rerouted lemmas provided by tokenizer"
"FastQT_Lemmatizer"	64	"Term rerouted to lemma field (wildcard lemmatization)"
"Final Query"	(none)	(none)

3.1.4.1.2.2.1.1.3   Refinement Section

The query refinement properties (the **rprop** element in the **refinement-section** element) provide information that enables the enhanced navigation of search results. The query refinement properties are described in the following table.

Refinement property name	Description
"NAME"	The formal name of the refiner.
"DNAM"	A human-readable, or display, name for the refiner.
"UCNT"	The number of used, or considered, hits for the refiner.
"TYPE"	The data type used by the refiner.

Refinement property name	Description
"UNIT"	The unit for the refiner—that is, the value of the **unit** attribute in the index schema.
"MODI"	The name of the managed property that is associated with the refiner.
"SCOR"	The score of the refiner. The score reflects the refinement potential. The default value is the entropy value "ETPY".
"HCNT"	The number of unique matching items for which the refiner has an observed value.
"SCNT"	The number of samples used. This value differs from that of the "HCNT" property because each item is counted once for each value that it has for the managed property being considered.
"RATI"	The ratio between the hit count and the number of documents retrieved from the search engine and used for computing the histogram over the bins.
"MIN"	The minimum value that is represented by the refiner. This value is applicable only to numeric and date/time refiners.
"MAX"	The maximum value that is represented by the refiner. This value is applicable only to numeric and date/time refiners.
"MEAN"	The average, or mean, value that is represented by the refiner. This value is applicable only to numeric and date/time refiners.
"ETPY"	The distribution, or entropy, of the results over the refinement bins. If all the results end up in one bin, the value of this property MUST be 0. The value MUST increase as items become more evenly distributed across the refinement bins. The value MUST NOT exceed the maximum of the logarithm of B, where B is the number of bins. Uniform distribution MUST result in the maximum value.
"SUM"	The sum of all the values for a numeric refiner across the result set.

The refinement properties in the following table represent parallel lists. Each list MUST be delimited by the value of **sep-list-value** and MUST contain the same number of entries. The entries MUST be related to each other by position in the list.

Refinement property name	Description
"NAMES"	The refinement names.
"MODS"	A list of refinement bins that are delimited by **list-entry-separator** characters.
"CNTS"	The refinement item counts.

3.1.4.1.2.2.1.1.4 Item Section

The following table describes the **ish-prop** item section header properties of the **item-section-header** element.

Name of item section header property	Description
"FIR"	The index of the first hit in the search result.
"LAS"	The index of the last hit in the search result.
"HTS"	The number of items that are included in the search result.
"CNT"	The total number of items that resulted from the query.
"TIM"	The amount of time, in seconds, that was used to process the request.
"MAR"	The theoretical maximum rank of any item that matches the request.

Each **item** element MUST contain the **item-property** item properties that are described in the following table.

Name of item property	Description
"rank"	The rank value of the item.
"fcoid"	The numeric value of the field used for collapsing.
"fcocount"	The number of collapsed items that are represented by this item.
"morehits"	A value that MUST be 1 if the item represents additional collapsed items—that is, if the value of "fcocount" is greater than or equal to 1.

Each **item** element MUST contain an **item-managed-property** element for each managed property that is specified as a member of the **summary class** named "servedcontent", as specified in [MS-FSSCFG] section 2.18.

3.1.4.1.2.2.1.1.5   Paging Section

The paging section contains implementation-specific data that MUST be discarded.

3.1.4.1.2.2.1.2   No Items

If no items exist in the search result, the search result item section MUST contain a **no-items** element, as specified in section 3.1.4.1.2.2.1.1. This condition occurs either because no items match the query expression or because the matches have been excluded from the search result by other search parameters, as specified in section 3.1.4.1.2.1.1.

The value of the **total-match-count** element MUST be the total number of items that match the query expression.

The value of the **query-time** element MUST be the amount of time, in seconds, that elapsed while the search was being processed.

3.1.4.1.2.2.1.3   Error

If the query processing component cannot complete a search, it MUST provide a search result that is formatted according to the following ABNF rules:

```
search-error = segment-name separator
    primary-error *secondary-error *CRLF

segment-name = "#SEG NAM webcluster" CRLF

primary-error = primary-error-code primary-error-text separator
primary-error-code = "#ERC " error-code-value CRLF
primary-error-text = "#ERT " error-text-value CRLF

secondary-error = secondary-error-code secondary-error-text separator
secondary-error-code = "#ERR COD " error-code-value CRLF
secondary-error-text = "#ERR TXT " error-text-value CRLF

error-code-value = unsigned-integer-value
error-text-value = string-value

unsigned-integer-value = 1*DIGIT
string-value = *(WSP/VCHAR/ichar)

separator = "####" CRLF

; <ichar> can be any international character (not US-ASCII).
ichar = %x7f-ffffffff
```

The **error-text-value** element contains descriptive text about the error and MUST NOT be processed.

Values for the **error-code-value** element that are less than 1000 correspond to HTTP status codes as specified in [RFC2616] section 10. The following table lists the additional valid **error-code-value** values.

Value	Description
1001	An unexpected internal error occurred.
1002	The query processing component could not process the query. This error does not apply to syntax errors in FQL expressions as specified in [MS-FSFQL]. An FQL syntax error MUST be reported with an **error-code-value** of 1201.
1003	All the **index partitions** are unavailable.
1005	The **search service application** is overloaded. The protocol client SHOULD resubmit the query.
1006	The requested functionality is not implemented.
1007	The query processing component did not finish the query because there were insufficient resources. The protocol client SHOULD resubmit the query.
1008	The connection to one or more **query matching nodes** was lost. The protocol client SHOULD resubmit the query.
1009	Multiple errors occurred from different index partitions. The protocol client SHOULD resubmit the query.
1010	An internal error occurred while the query was being evaluated.
1011	The query timed out while waiting for responses from query matching nodes. The time spent executing the query exceeded the specified maximum. The protocol client SHOULD resubmit the

Value	Description
	query.
1012	The query is too complex. The query processing component ran out of resources while attempting to process the query.
1013	The query processing component could not finish the query because internal resources were temporarily unavailable. The protocol client MAY resubmit the query.
1014	The syntax of the query is not supported.
1015	The query processing component did not perform the search because it was unable to retrieve a valid license.
1016	The search index is unavailable.
1017	The query expression contains more wildcard characters than are permitted.
1018	One or more index partitions are unavailable.
1020	An internal error occurred while a document summary was being generated. The protocol client SHOULD resubmit the query.
1021	A severe internal error occurred while a document summary was being retrieved or generated.
1022	A timeout occurred while a document summary was being retrieved or generated.
1101	The query did not specify any search criteria.
1102	The query processing component could not initiate communication with a query matching node.
1103	No query was specified.
1104	The query processing component could not send the query to a query matching node.
1105	The query timed out for an unspecified reason. The time spent executing the query exceeded the specified maximum. The protocol client SHOULD resubmit the query.
1106	The query processing component received an unknown response while waiting for search results.
1107	A communication error occurred while the query processing component was attempting to send the query to a query matching node.
1108	An error occurred while the query processing component was requesting a document summary.
1109	The retrieval of document summaries took longer than the maximum amount of time permitted for the operation.
1110	An error occurred while the query processing component was attempting to establish a connection to retrieve document summaries.
1111	The query processing component received an invalid response while fetching document summaries.
1112	The query processing component failed to store search result information.
1113	The query processing component failed to allocate memory for the query.
1114	An error occurred while the query processing component was initiating communication with one or more query matching nodes. The query processing component was unable to retrieve a complete set of results.

Value	Description
1201	An error occurred while the query processing component was parsing the FQL expression in the query.
1202	An error occurred during the processing of search results.
1999	An unspecified error occurred in the query processing component.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4   Protocol Examples

4.1   ProcessRequest Examples

This section contains examples for the **request** element as specified in section 3.1.4.1.2.1.

4.1.1   Query Example

The following example searches for items that contain the word *cat*:

```
http://q/cgi-bin/search?query=cat
```

4.1.2   Query Type Examples

The following example uses SharePoint Search Keyword Syntax to search for items that contain both the word *cat* and the word *dog*:

```
http://q/cgi-bin/search?query=cat%20dog&type=kwall
```

The following example uses FQL to search for items that contain both the word *cat* and the word *dog*:

```
http://q/cgi-bin/search?query=string(%22cat%20dog%22%2C%20mode%3D%22and%22)
```

4.1.3   Filter Example

The following example searches for the word *butter* in items that have a managed property named **title** for which the value is "Recipe for a Baked Potato":

```
http://q/cgi-
bin/search?query=butter&qtf_parsekw:filter=title:equals(%22Recipe%20for%20a%20Baked%20Potato%
22)&type=kwall
```

4.1.4   Freshness Boost Example

The following example boosts items in the search result based on their ages relative to August 27, 2003 at 20:16:00 **Coordinated Universal Time (UTC)**:

```
http://q/cgi-bin/search?query=cat&qtf_freshnessboost:datetime=2003-08-27T20:16:00
```

4.1.5   Similarity Example

The following example returns items that contain the word *fruit*. Items that also include the word *pear* are considered more relevant than those that do not include *pear* but do include *kiwi*. Items that include neither *pear* nor *kiwi* are considered the least relevant.

```
http://q/cgi-bin/search?query=fruit&similarto=[pear,0.9][kiwi,0.7]&similartype=find
```

4.1.6 Sorting Example

The following example sorts the results by the managed property named **price** in ascending order:

```
http://q/cgi-bin/search?query=laptop&sortby=price
```

The following example sorts the results by the managed property named **price** in ascending numeric order and then by the rank profile named **fresh**:

```
http://q/cgi-bin/search?query=laptop&sortby=+price%20fresh
```

The following example randomly sorts the search result:

```
http://q/cgi-bin/search?query=airplanes&sortby=-[rank]
+[random:seed=5432:hashfield=hashField]
```

The following example assigns a random value from 0 through 200 to each item's rank, thus achieving a degree of randomization:

```
http://q/cgi-bin/search?query=airplanes&sortby=-[rank]
+[random:seed=5432:addtorankmax=200]&sortdirection=ascending
```

4.1.7 Stemming Example

The following example turns on stemming and typically matches items containing either the word *airplane* or the word *airplanes*:

```
http://q/cgi-bin/search?query=airplane&qtf_lemmatize=on
```

4.1.8 Spell-Checking Example

In the following example, the misspelled word *thesarus* will be corrected to *thesaurus* before being processed:

```
http://q/cgi-bin/search?query=thesarus&spell=on
```

4.1.9 Automatic Resubmission Example

The following example enables spell-checking in **suggest** mode:

```
http://q/cgi-bin/search?query=thesarus&resubmitflags=16176
```

4.1.10 Query Refinement Example

The following example requests refiners based on the **author**, **integer**, and **double** managed properties:

```
http://q/cgi-bin/search?
&query=beans&rpf_navigation:enabled=1&rpf_navigation:navigators=author%2Cinteger%2Cdouble
```

4.1.11   Field Collapsing Example

The following example collapses all similar items based on size:

```
http://q/cgi-
bin/search?query=%22Adventure%20Works%22%20%22Graphic%20Design%20Institute%22%20%22Southridge
%20Video%22&collapseon=batvsize
```

4.1.12   Dynamic Duplicate Removal Example

The following example defines the duplicates as the items that have the same value for the **title** managed property and the same value for the **body** managed property. Any items in the search result that have the same **title** and the same **body** are removed from the search result, leaving only the first occurrence.

```
http://q/cgi-bin/search?query=dog&rff_ddr:enabled=1 rff_ddr:slot1=bsumtitle,bsumbody
```

4.1.13   Query Language Example

The following example sets the query string language to German ("de") to apply to language-specific query processing, such as spell-checking and stemming.

```
http://q/cgi-bin/search?query=fast%20goethe&type=kwall&language=de
```

4.1.14   White Space Preservation Example

The following example turns on white space preservation, thus ensuring that newline characters, tabs, and so forth will be returned in the search result.

```
http://q/cgi-bin/search?query=cat&tvmarg9=1
```

4.1.15   Limiting Results Example

The following example sets the maximum number of hits in the search result to 100:

```
http://q/cgi-bin/search?query=cat&hits=100
```

4.1.16   Skipping Results Example

The following example returns the second page of results for items containing the word *potato*:

```
http://q/cgi-bin/search?query=potato&type=kwall&offset=10&hits=10
```

4.2 ProcessRequestResponse Examples

4.2.1 Query Transform Example

The following example shows the query transform section of the result for a query transform named **FastQT_Keyword**:

```
####
#QTF NAM FastQT_Keyword
#QTF ACT Suggested new query
#QTF QRY fire
#QTF CUS <KeywordData />
#QTF MSG Keyword processing
#QTF MID 1
####
```

4.2.2 Refinement Example

The following example shows a query that specifies three refiners:

```
http://q/cgi-bin/search?
query=beans&rpf_navigation:enabled=1&rpf_navigation:navigators=author%2Cinteger%2Cfloat
```

The query processing component returns three refiners, each of a different type—**String**, **Float**, or **Integer**—in the search result:

```
####
#NAV ENT 3
####
#NAV NAME author
#NAV DNAM author
#NAV TYPE String
#NAV UNIT
#NAV MODI author
#NAV SCOR 2.988613844
#NAV UCNT 196
#NAV HCNT 196
#NAV SCNT 196
#NAV RATI 1
#NAV MIN  0
#NAV MAX  0
#NAV MEAN 0
#NAV ETPY 2.988613962
#NAV SUM  0
#NAV NAMES
Lauren~Peter~Irene~Thomas~Adrienne~Charlie~David~Frank~Greg~James~Michael~Nancy~Scott~Wallace
~Karen~Quincy~Bill~Elise~Henry~Veronica
#NAV MODS
^Lauren$~^Peter$~^Irene$~^Thomas$~^Adrienne$~^Charlie$~^David$~^Frank$~^Greg$~^James$~^Michae
l$~^Nancy$~^Scott$~^Wallace$~^Karen$~^Quincy$~^Bill$~^Elise$~^Henry$~^Veronica$
#NAV CNTS 12~12~11~11~10~10~10~10~10~10~10~10~10~9~9~8~8~8~8
####
#NAV NAME float
#NAV DNAM float
#NAV TYPE Float
#NAV UNIT
```

```
#NAV MODI float
#NAV SCOR 1.386294365
#NAV UCNT 196
#NAV HCNT 196
#NAV SCNT 196
#NAV RATI 1
#NAV MIN  9.80000000000000700000000000000000000000000000000000000000000000
#NAV MAX  396.69999999999999000000000000000000000000000000000000000000000000
#NAV MEAN 203.25
#NAV ETPY 1.386294361
#NAV SUM  39837
#NAV NAMES auto~From 103 to 203~From 203 to 303~303 and above
#NAV MODS [;103>~[103;203>~[203;303>~[303;>
#NAV CNTS 49494949
####
#NAV NAME integer
#NAV DNAM integer
#NAV TYPE Integer
#NAV UNIT
#NAV MODI integer
#NAV SCOR 1.386294365
#NAV UCNT 196
#NAV HCNT 196
#NAV SCNT 196
#NAV RATI 1
#NAV MIN  89
#NAV MAX  3958
#NAV MEAN 2023.5
#NAV ETPY 1.386294361
#NAV SUM  396606
#NAV NAMES auto~From 1022 to 2022~From 2023 to 3023~3024 and above
#NAV MODS [;1021]~[1022;2022]~[2023;3023]~[3024;]
#NAV CNTS 49~49~49~49
####
```

4.2.3 Item Section Example

The following example shows the item section header of the search result:

```
####
#FIR 1
#LAS 10
#HTS 10
#CNT 42
#TIM 0.1700
#MAR 2754
###/
```

4.2.4 Multiple Items Example

The following example shows a query that searches for the word *frisky*:

```
http://q/cgi-bin/search?offset=0&hits=10&query=frisky
```

Here is the search result of the preceding query:

```
#SEG NAM webcluster
#MTA SEP ~
####
#QTF NAM Original query
#QTF ACT NOP
#QTF QRY
and(filter(or(docacl:all,docacl:unknown),annotation_class="invisible",annotation_class="sam")
,frisky)
####
#QTF NAM FastQT_Keyword
#QTF ACT Suggested new query
#QTF QRY frisky
#QTF CUS <KeywordData />
#QTF MSG Keyword processing
#QTF MID 1
####
#QTF NAM FastQT_Lemmatizer
#QTF ACT nop
#QTF QRY
#QTF CUS No change to query
#QTF MSG Lemmatization turned off for current query
#QTF MID 16
####
#QTF NAM Final query
#QTF ACT NOP
#QTF QRY and(filter(or("docacl":string("all"), "docacl":string("unknown")),
annotation_class="invisible", annotation_class="sam"), string("frisky"))
#QTF CUS FQL
####
#NAV ENT 0
####
#FIR 1
#LAS 2
#HTS 2
#CNT 2
#TIM 0.0150
#MAR 5201
###/
#### 1
#rank 762
#ranklog
#fcoid 0
#fcocount 0
#morehits 0
#internalid 59e10aa5aec1d8bc6b246c6648af6d18_sp
#contentid http://dummy.com/cat.txt
#contentids http://dummy.com/cat.txt
#collection sp
#title cat.txt
#body  Cats are sometimes <b>frisky</b>, but often are lethargic.
#teaser   Cats are sometimes frisky, but often are lethargic.
#contenttype text/plain
#format Text
#language en
#languages en
#charset iso-8859-1
#urls http://dummy.com/cat.txt
#url http://dummy.com/cat.txt
#domain
```

```
#tld
#path /cat.txt
#crawltime
#processingtime 2009-09-10T15:01:07Z
#docdatetime 2009-09-10T15:01:11Z
#size 53
#docvector
#documentsignature 166509901429199529
#hwboost
#docrank 0
#siterank 0
#urldepthrank 630
#docacl unknown
#docaclsystemid unknown
#author
#createdby
#fileextension TXT
#isdocument
#modifiedby
#account
#assignedto
#doccomments
#dockeywords
#spdocid
#docsubject
#created
#lastmodifiedtime
#notes
#siteid
#sitename
#sitetitle
#spsiteurl
#status
#crawledpropertynames
#detectedlanguage en
#companies
#locations
#personnames
#concepts
#taxonomy
#companyteaser
#locationteaser
#personnameteaser
#owsmetadatafacetinfo
#xml
###/
#### 2
#rank 762
#ranklog
#fcoid 0
#fcocount 0
#morehits 0
#internalid b83fb55f67e2d5f21cfc81ffc6bb5359_sp
#contentid http://dummy.com/dog.txt
#contentids http://dummy.com/dog.txt
#collection sp
#title dog.txt
#body  Dogs are often <b>frisky</b>, and usually smell funny.
#teaser   Dogs are often frisky, and usually smell funny.
```

```
#contenttype text/plain
#format Text
#language en
#languages en
#charset iso-8859-1
#urls http://dummy.com/dog.txt
#url http://dummy.com/dog.txt
#domain
#tld
#path /dog.txt
#crawltime
#processingtime 2009-09-10T15:01:24Z
#docdatetime 2009-09-10T15:01:25Z
#size 49
#docvector
#documentsignature 145710094476283743
#hwboost
#docrank 0
#siterank 0
#urldepthrank 630
#docacl unknown
#docaclsystemid unknown
#author
#createdby
#fileextension TXT
#isdocument
#modifiedby
#account
#assignedto
#doccomments
#dockeywords
#spdocid
#docsubject
#created
#lastmodifiedtime
#notes
#siteid
#sitename
#sitetitle
#spsiteurl
#status
#crawledpropertynames
#detectedlanguage en
#companies
#locations
#personnames
#concepts
#taxonomy
#companyteaser
#locationteaser
#personnameteaser
#owsmetadatafacetinfo
#xml
###/
```

4.2.5 No Items Example

The following example shows a query that searches for the word *monkey*:

```
http://q/cgi-bin/search?offset=0&hits=10&query=monkey
```

Here is the search result of the preceding query, which matches no items:

```
#SEG NAM webcluster
#MTA SEP ~
####
#QTF NAM Original query
#QTF ACT NOP
#QTF QRY
and(filter(or(docacl:all,docacl:unknown),annotation_class="invisible",annotation_class="sam")
,monkey)
####
#QTF NAM FastQT_Keyword
#QTF ACT Suggested new query
#QTF QRY monkey
#QTF CUS <KeywordData />
#QTF MSG Keyword processing
#QTF MID 1
####
#QTF NAM FastQT_Lemmatizer
#QTF ACT nop
#QTF QRY
#QTF CUS No change to query
#QTF MSG Lemmatization turned off for current query
#QTF MID 16
####
#QTF NAM FastQT_ResubmitQuery
#QTF ACT nop
#QTF QRY
#QTF CUS Resubmit off flag present, skipping resubmit.
#QTF MSG Query not resubmitted
#QTF MID 2
####
#QTF NAM Final query
#QTF ACT NOP
#QTF QRY and(filter(or("docacl":string("all"), "docacl":string("unknown")),
annotation_class="invisible", annotation_class="sam"), string("monkey"))
#QTF CUS FQL
####
#NAV ENT 0
####
#FIR 1
#LAS 0
#HTS 0
#CNT 0
#TIM 0.0030
#MAR 0
###/
#C 0
#T 0.0030
```

4.2.6 Error Example

The following example shows an invalid query:

```
http://q/cgi-bin/search?offset=0&hits=10&query=zebra%20stripes
```

Here is the search result of the preceding query:

```
#SEG NAM webcluster
####
#ERC 1201
#ERT parsefql: Query Error: line 1:104: unexpected token: stripes
####
```

5   Security

5.1   Security Considerations for Implementers

Items in search results can contain sensitive information, so it is important to include in a search result only those items that the identified user is authorized to view. The FAST Search Authorization (FSA) worker component performs this function using information that is synchronized by the FSA manager as described in [MS-FSSAS].

5.2   Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the following full WSDL is provided:

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:tns="http://Microsoft.SharePoint.Search.Extended.QRServer"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://Microsoft.SharePoint.Search.Extended.QRServer"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema xmlns:tns="http://Microsoft.SharePoint.Search.Extended.QRServer"
elementFormDefault="qualified"
targetNamespace="http://Microsoft.SharePoint.Search.Extended.QRServer">
      <xs:simpleType name="StreamBody">
        <xs:restriction base="xs:base64Binary" />
      </xs:simpleType>
      <xs:element name="ProcessRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="request" nillable="true" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ProcessRequestResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProcessRequestResult" type="tns:StreamBody" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="IProxyRemote_ProcessRequest_InputMessage">
    <wsdl:part name="parameters" element="tns:ProcessRequest" />
  </wsdl:message>
  <wsdl:message name="IProxyRemote_ProcessRequest_OutputMessage">
    <wsdl:part name="parameters" element="tns:ProcessRequestResponse" />
  </wsdl:message>
  <wsdl:portType name="IProxyRemote">
    <wsdl:operation name="ProcessRequest">
      <wsdl:input
wsaw:Action="http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequest
" message="tns:IProxyRemote_ProcessRequest_InputMessage" />
      <wsdl:output
wsaw:Action="http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequest
Response" message="tns:IProxyRemote_ProcessRequest_OutputMessage" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding_IProxyRemote" type="tns:IProxyRemote">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ProcessRequest">
      <soap:operation
soapAction="http://Microsoft.SharePoint.Search.Extended.QRServer/IProxyRemote/ProcessRequest"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
```

```
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  </wsdl:definitions>
```

7   Appendix B: Product Behavior

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

▪ Microsoft® FAST™ Search Server 2010

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

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

8   Change Tracking

This section identifies changes that were made to the [MS-FSQR] protocol document between the November 2010 and December 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.

- An extensive rewrite, addition, or deletion of major portions of content.

- Changes made for template compliance.

- Removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- Obsolete document removed.

Editorial changes are always classified with the change type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
7 Appendix B: Product Behavior	Updated the list of applicable product versions.	N	Content updated.

9   Index

A

Release: Sunday, December 19, 2010