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

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/06/2009</td>
<td>0.1</td>
<td>Major</td>
<td>Initial Availability</td>
</tr>
<tr>
<td>02/19/2010</td>
<td>1.0</td>
<td>Minor</td>
<td>Updated the technical content</td>
</tr>
<tr>
<td>03/31/2010</td>
<td>1.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>04/30/2010</td>
<td>1.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>06/07/2010</td>
<td>1.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>06/29/2010</td>
<td>1.04</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>07/23/2010</td>
<td>1.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>09/27/2010</td>
<td>1.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/15/2010</td>
<td>1.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>12/17/2010</td>
<td>1.05</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>03/18/2011</td>
<td>1.05</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>06/10/2011</td>
<td>1.05</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>01/20/2012</td>
<td>1.6</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>04/11/2012</td>
<td>1.6</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>07/16/2012</td>
<td>1.6</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction .................................................................................................................. 6
  1.1 Glossary .................................................................................................................. 6
  1.2 References .............................................................................................................. 7
    1.2.1 Normative References ....................................................................................... 7
    1.2.2 Informative References ..................................................................................... 8
  1.3 Protocol Overview (Synopsis) .................................................................................. 8
  1.4 Relationship to Other Protocols ............................................................................. 8
  1.5 Prerequisites/Preconditions ..................................................................................... 9
  1.6 Applicability Statement ......................................................................................... 9
  1.7 Versioning and Capability Negotiation ................................................................. 9
  1.8 Vendor-Extensible Fields ....................................................................................... 9
  1.9 Standards Assignments ......................................................................................... 9

2 Messages ...................................................................................................................... 10
  2.1 Transport .................................................................................................................. 10
  2.2 Common Message Syntax ....................................................................................... 10
    2.2.1 Namespaces ......................................................................................................... 10
    2.2.2 Messages ............................................................................................................. 10
    2.2.3 Elements ............................................................................................................. 10
    2.2.4 Complex Types .................................................................................................. 10
    2.2.5 Simple Types ...................................................................................................... 10
    2.2.6 Attributes .......................................................................................................... 10
    2.2.7 Groups ............................................................................................................... 10
    2.2.8 Attribute Groups ............................................................................................... 10

3 Protocol Details ......................................................................................................... 11
  3.1 Query and Result Protocol Server Details ............................................................. 11
    3.1.1 Abstract Data Model .......................................................................................... 11
    3.1.2 Timers ............................................................................................................... 11
    3.1.3 Initialization ........................................................................................................ 11
    3.1.4 Message Processing Events and Sequencing Rules .......................................... 11
      3.1.4.1 ProcessRequest ............................................................................................. 11
        3.1.4.1.1 Messages ................................................................................................. 12
          3.1.4.1.1.1 IProxyRemote_ProcessRequest_InputMessage .................................. 12
          3.1.4.1.1.2 IProxyRemote_ProcessRequest_OutputMessage ................................ 12
        3.1.4.1.2 Elements ................................................................................................. 12
          3.1.4.1.2.1 ProcessRequest ................................................................................. 12
            3.1.4.1.2.1.1 Search Query ................................................................................. 13
              3.1.4.1.2.1.1.1 Query Parameter ..................................................................... 13
              3.1.4.1.2.1.1.2 Query Type Parameter ............................................................ 13
              3.1.4.1.2.1.1.3 Filter Parameter ....................................................................... 13
              3.1.4.1.2.1.1.4 Freshness Boost Parameter ....................................................... 14
              3.1.4.1.2.1.1.5 Similarity Parameters ............................................................... 14
            3.1.4.1.2.1.1.6 Sorting Parameters ................................................................. 15
              3.1.4.1.2.1.1.6.1 sortby Parameter ................................................................. 15
              3.1.4.1.2.1.1.7 Stemming Parameter ............................................................... 18
              3.1.4.1.2.1.1.8 Spell-Checking Parameter ......................................................... 18
              3.1.4.1.2.1.1.9 Automatic Resubmission Parameter ....................................... 18
            3.1.4.1.2.1.1.10 Query Refinement Parameters .............................................. 19
            3.1.4.1.2.1.1.11 Field Collapsing Parameters ............................................... 20
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.

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]:

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
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
Simple Object Access Protocol (SOAP)
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

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.


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

[MS-FSFL] Microsoft Corporation, "Fast Query Language Structure".


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

Queries that use this protocol are expressed in either SharePoint Search Keyword Syntax, as described in [MS-SEARCH] section 2.2.10, 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 3.1.4.1.2.1. To control how the request will be processed and to generate the query request, as specified in [MS-FSDQF] section 2.2.6, the protocol server MUST evaluate the parameters as specified in section 3.1.4.1.2.1.

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.

```xml
<wsdl:operation name="ProcessRequest">
  <wsdl:input
    message="tns:IProxyRemote_ProcessRequest_InputMessage" />
  <wsdl:output
    message="tns:IProxyRemote_ProcessRequest_OutputMessage" />
</wsdl:operation>
```

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:

```
```

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:

```
```

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.

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

**request:** A valid search query, as specified in section **3.1.4.1.2.1**.
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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;kwany&quot;</td>
<td>The query parameter MUST be expressed in SharePoint Search Keyword Syntax, as specified in [MS-SEARCH] section 2.2.10. For an item to be included in the search result, at least one of the keywords from the query MUST be matched.</td>
</tr>
<tr>
<td>&quot;kwall&quot;</td>
<td>The query parameter MUST be expressed in SharePoint Search Keyword Syntax, as specified in [MS-SEARCH] section 2.2.10. For an item to be included in the search result, all the keywords of the query MUST be matched.</td>
</tr>
</tbody>
</table>

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".

---

[MS-FSQR] — v20120630
Query and Result Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: July 16, 2012
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 = ("0" %x30-32 DIGIT) ; 00-29
    / ("3" %x30-31) ; 30-31 (00-31) - two-digit day
hour = ("0" %x30-31 DIGIT) ; 00-19
    / ("2" %x30-33) ; 20-23 (00-23) - two-digit hour
minute = ("0" %x30-35 DIGIT) ; 00-59 - two-digit minute
second = ("0" %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.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpf_sortsimilar:enabled</td>
<td>Flag</td>
<td>Specifies whether to use document vectors to refine dynamic ranks. A value of 1 means yes, and a value of 0 means no.</td>
</tr>
<tr>
<td>similartype</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>similarto</td>
<td>String</td>
<td>Specifies one or more document vectors, each in the form <code>[term,weight]</code>, 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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;find&quot;</td>
<td>Includes items that match the query, the similarto parameter, or both. That is, the similarto parameter acts as an OR condition to the query.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sortdirection</td>
<td>String</td>
<td>Specifies the default sort direction. The valid values for this parameter are described later in this section.</td>
</tr>
<tr>
<td>sortby</td>
<td>String</td>
<td>Specifies the criteria for sorting, as specified in section 3.1.4.1.2.1.1.6.1.</td>
</tr>
<tr>
<td>random</td>
<td>Integer</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;ascending&quot;</td>
<td>Sorts in ascending order (smallest first, largest last).</td>
</tr>
<tr>
<td>&quot;descending&quot;</td>
<td>Sorts in descending order (largest first, smallest last). This value acts as the default.</td>
</tr>
</tbody>
</table>

#### 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 = "-"
```

[MS-FSQR] — v20120630
Query and Result Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: July 16, 2012
random-seed = unsigned-integer
hashfield = managed-property
addtorankmax = integer
integer = ["-" / "+"] 1*DIGIT
unsigned-integer = 1*DIGIT

equation = (function / managed-property / rank
    / parenthetic-equation / number)
    *(operator equation)
parenthetic-equation = "(" equation ")"
function = unary-function / pow / atan2 / bin
unary-function = ("sqrt" / "exp" / "log" / "abs" / "ceil" / "floor"
    / "round" / "sin" / "cos" / "tan" / "asin" / "acos" / "atan")
    "(" equation ")"
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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>managed-property</td>
<td>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 &quot;Enabled&quot; as specified in [MS-FSSCFG] section 1.3.2.1.</td>
</tr>
<tr>
<td>rank-profile</td>
<td>The items in the search result MUST be sorted by the specified rank profile.</td>
</tr>
<tr>
<td>rank</td>
<td>The items in the search result MUST be sorted by dynamic rank.</td>
</tr>
<tr>
<td>random</td>
<td>The items in the search result MUST exist in pseudo-random order.</td>
</tr>
<tr>
<td>formula</td>
<td>The items in the search result MUST be sorted by the numeric result of the specified expression element.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>random-seed</td>
<td>The integer seed to use when generating the pseudo-random order for the search result.</td>
</tr>
<tr>
<td>hashfield</td>
<td>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.</td>
</tr>
</tbody>
</table>
Element | Description
--- | ---
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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqrt(n)</td>
<td>The square root of n.</td>
</tr>
<tr>
<td>exp(n)</td>
<td>The exponential function that is equivalent to $\text{pow}(2.71828182846,n)$.</td>
</tr>
<tr>
<td>log(n)</td>
<td>The natural logarithm of n.</td>
</tr>
<tr>
<td>abs(n)</td>
<td>The absolute value of n.</td>
</tr>
<tr>
<td>ceil(n)</td>
<td>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.</td>
</tr>
<tr>
<td>floor(n)</td>
<td>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.</td>
</tr>
<tr>
<td>round(n)</td>
<td>The rounding of n to the nearest whole number.</td>
</tr>
<tr>
<td>sin(n)</td>
<td>The sine of n radians.</td>
</tr>
<tr>
<td>cos(n)</td>
<td>The cosine of n radians.</td>
</tr>
<tr>
<td>tan(n)</td>
<td>The tangent of n radians.</td>
</tr>
<tr>
<td>asin(n)</td>
<td>The arcsine, in radians, of n.</td>
</tr>
<tr>
<td>acos(n)</td>
<td>The arccosine, in radians, of n.</td>
</tr>
<tr>
<td>atan(n)</td>
<td>The arctangent, in radians, of n.</td>
</tr>
<tr>
<td>pow(x,y)</td>
<td>The value of x raised to the power of y.</td>
</tr>
<tr>
<td>atan2(y,x)</td>
<td>A two-argument arctangent—the angle in radians between the positive x axis and the specified Cartesian coordinate (x,y).</td>
</tr>
<tr>
<td>bucket(b,n1,...)</td>
<td>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...</td>
</tr>
</tbody>
</table>
### Function Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>highest bin.</td>
</tr>
</tbody>
</table>

#### 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;on&quot;</td>
<td>Enables spell-checking and correction for the query</td>
</tr>
<tr>
<td>&quot;suggest&quot;</td>
<td>Specifies that the result will contain the corrected query but that the query will be executed without correction</td>
</tr>
<tr>
<td>&quot;off&quot;</td>
<td>Disables spell-checking and correction</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Do not resubmit.</td>
</tr>
<tr>
<td>64</td>
<td>Resubmit with stemming enabled.</td>
</tr>
<tr>
<td>7984</td>
<td>Resubmit with spell-checking enabled.</td>
</tr>
<tr>
<td>8192</td>
<td>Resubmit with spell-checking in <code>suggest</code> 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 <code>AND</code> operator, which results in 16176.</td>
</tr>
<tr>
<td>134217728</td>
<td>Resubmit with synonyms added.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpf_navigation:enabled</td>
<td>Flag</td>
<td>Whether query refinement is enabled for the query.</td>
</tr>
<tr>
<td>rpf_navigation:navigators</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>rpf_navigation:hits</td>
<td>Integer</td>
<td>The number of items to use when creating refiners.</td>
</tr>
</tbody>
</table>

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 = "*" / (1*bin-name-char)
bin-name-char = ALPHA / DIGIT / WSP
deehpits = "deehpits=" 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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max-bins</td>
<td>The refiner MUST NOT return more refinement bins than the value of max-bins.</td>
</tr>
<tr>
<td>min-frequency</td>
<td>The refiner MUST NOT return any refinement bin that has a frequency value less than that of min-frequency.</td>
</tr>
<tr>
<td>bin-name-prefix</td>
<td>The refiner MUST NOT return any refinement bin unless its name begins with the value of bin-name-prefix.</td>
</tr>
</tbody>
</table>

The options for the sort element are described in the following table.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;frequency&quot;</td>
<td>The refiner MUST sort its refinement bins by number of items.</td>
</tr>
<tr>
<td>&quot;name&quot;</td>
<td>The refiner MUST sort its refinement bins by name.</td>
</tr>
<tr>
<td>&quot;number&quot;</td>
<td>The refiner MUST sort numerically by refinement bin name. For example, the name &quot;05&quot; MUST be considered greater than the name &quot;4&quot;.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min-frequency</td>
<td>The refiner MUST NOT return refinement bins with frequency values less than that of min-frequency.</td>
</tr>
<tr>
<td>min-bins</td>
<td>The refiner MUST NOT return fewer than refinement bins than the value of min-bins.</td>
</tr>
<tr>
<td>max-bins</td>
<td>The refiner MUST NOT return more refinement bins than the value of max-bins.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collapseon</td>
<td>String</td>
<td>The managed property name to collapse. The managed property name MUST be prefixed with &quot;batv&quot;.</td>
</tr>
<tr>
<td>collapsenum</td>
<td>Integer</td>
<td>The number of items to include from the collapsing process. The default value is 1.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rff_ddr:enabled</td>
<td>Flag</td>
<td>Whether to enable or disable dynamic duplicate removal.</td>
</tr>
<tr>
<td>rff_ddr:slot1</td>
<td>String</td>
<td>The primary slot for specifying managed properties that the query processing component uses for dynamic duplicate removal.</td>
</tr>
<tr>
<td>rff_ddr:slot2</td>
<td>String</td>
<td>The secondary slot for dynamic duplicate removal criteria.</td>
</tr>
</tbody>
</table>

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

qtp-body = *qt-property
qt-property = qtp-string / qtp-msg
qtp-string = qtp-prefix qt-string-prop-name [vwsp string-value] CRLF
qtp-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-header *{item-property / item-managed-property}
item-property = "#" ("rank" / "fcoid" / "fcocount" / "morehits")
vwsp unsigned-integer-value CRLF

item-managed-property = "###" 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-start-index vwsp page-end-index vwsp page-query
page-start-index = unsigned-integer-value
page-end-index = 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)
/ (\%x30-31) ; two-digit day (30-31)
hour = (\%x30-31 DIGIT) ; two-digit hour (00-19)
/ (\%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 = %x?f-fffffff

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

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.
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.

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

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.

<table>
<thead>
<tr>
<th>Value of &quot;NAM&quot; property</th>
<th>Value of &quot;ACT&quot; property</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Original query&quot;</td>
<td>&quot;NOP&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Keyword&quot;</td>
<td>&quot;Suggested new query&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>&quot;nop&quot;</td>
</tr>
<tr>
<td>&quot;Final Query&quot;</td>
<td>&quot;NOP&quot;</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value of &quot;NAM&quot; property</th>
<th>&quot;MID&quot; value</th>
<th>&quot;MSG&quot; value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Original query&quot;</td>
<td>(none)</td>
<td>(none)</td>
</tr>
<tr>
<td>&quot;FastQT_Keyword&quot;</td>
<td>1</td>
<td>&quot;Keyword processing&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>1</td>
<td>&quot;Lemmatized term (query reduction)&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>2</td>
<td>&quot;Lemmatized term (query expansion)&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>4</td>
<td>&quot;Term rerouted (lemmatization by document expansion)&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>8</td>
<td>&quot;No query transformation&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>16</td>
<td>&quot;Lemmatization turned off for current query&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>32</td>
<td>&quot;Rerouted lemmas provided by tokenizer&quot;</td>
</tr>
<tr>
<td>&quot;FastQT_Lemmatizer&quot;</td>
<td>64</td>
<td>&quot;Term rerouted to lemma field (wildcard lemmatization)&quot;</td>
</tr>
<tr>
<td>&quot;Final Query&quot;</td>
<td>(none)</td>
<td>(none)</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Refinement property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NAME&quot;</td>
<td>The formal name of the refiner.</td>
</tr>
<tr>
<td>&quot;DNAM&quot;</td>
<td>A human-readable, or display, name for the refiner.</td>
</tr>
<tr>
<td>&quot;UCNT&quot;</td>
<td>The number of used, or considered, hits for the refiner.</td>
</tr>
<tr>
<td>&quot;TYPE&quot;</td>
<td>The data type used by the refiner.</td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>Name of item section header property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;FIR&quot;</td>
<td>The index of the first hit in the search result.</td>
</tr>
<tr>
<td>&quot;LAS&quot;</td>
<td>The index of the last hit in the search result.</td>
</tr>
<tr>
<td>&quot;HTS&quot;</td>
<td>The number of items that are included in the search result.</td>
</tr>
<tr>
<td>&quot;CNT&quot;</td>
<td>The total number of items that resulted from the query.</td>
</tr>
<tr>
<td>&quot;TIM&quot;</td>
<td>The amount of time, in seconds, that was used to process the request.</td>
</tr>
<tr>
<td>&quot;MAR&quot;</td>
<td>The theoretical maximum rank of any item that matches the request.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name of item property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;rank&quot;</td>
<td>The rank value of the item.</td>
</tr>
<tr>
<td>&quot;fcoid&quot;</td>
<td>The numeric value of the field used for collapsing.</td>
</tr>
<tr>
<td>&quot;fcocount&quot;</td>
<td>The number of collapsed items that are represented by this item.</td>
</tr>
<tr>
<td>&quot;morehits&quot;</td>
<td>A value that MUST be 1 if the item represents additional collapsed items—that is, if the value of &quot;fcocount&quot; is greater than or equal to 1.</td>
</tr>
</tbody>
</table>

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.

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>An unexpected internal error occurred.</td>
</tr>
<tr>
<td>1002</td>
<td>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.</td>
</tr>
<tr>
<td>1003</td>
<td>All the index partitions are unavailable.</td>
</tr>
<tr>
<td>1005</td>
<td>The search service application is overloaded. The protocol client SHOULD resubmit the query.</td>
</tr>
<tr>
<td>1006</td>
<td>The requested functionality is not implemented.</td>
</tr>
<tr>
<td>1007</td>
<td>The query processing component did not finish the query because there were insufficient resources. The protocol client SHOULD resubmit the query.</td>
</tr>
<tr>
<td>1008</td>
<td>The connection to one or more query matching nodes was lost. The protocol client SHOULD resubmit the query.</td>
</tr>
<tr>
<td>1009</td>
<td>Multiple errors occurred from different index partitions. The protocol client SHOULD resubmit the query.</td>
</tr>
<tr>
<td>1010</td>
<td>An internal error occurred while the query was being evaluated.</td>
</tr>
<tr>
<td>1011</td>
<td>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 query.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1012</td>
<td>The query is too complex. The query processing component ran out of resources while attempting to process the query.</td>
</tr>
<tr>
<td>1013</td>
<td>The query processing component could not finish the query because internal resources were temporarily unavailable. The protocol client MAY resubmit the query.</td>
</tr>
<tr>
<td>1014</td>
<td>The syntax of the query is not supported.</td>
</tr>
<tr>
<td>1015</td>
<td>The query processing component did not perform the search because it was unable to retrieve a valid license.</td>
</tr>
<tr>
<td>1016</td>
<td>The search index is unavailable.</td>
</tr>
<tr>
<td>1017</td>
<td>The query expression contains more wildcard characters than are permitted.</td>
</tr>
<tr>
<td>1018</td>
<td>One or more index partitions are unavailable.</td>
</tr>
<tr>
<td>1020</td>
<td>An internal error occurred while a document summary was being generated. The protocol client SHOULD resubmit the query.</td>
</tr>
<tr>
<td>1021</td>
<td>A severe internal error occurred while a document summary was being retrieved or generated.</td>
</tr>
<tr>
<td>1022</td>
<td>A timeout occurred while a document summary was being retrieved or generated.</td>
</tr>
<tr>
<td>1101</td>
<td>The query did not specify any search criteria.</td>
</tr>
<tr>
<td>1102</td>
<td>The query processing component could not initiate communication with a query matching node.</td>
</tr>
<tr>
<td>1103</td>
<td>No query was specified.</td>
</tr>
<tr>
<td>1104</td>
<td>The query processing component could not send the query to a query matching node.</td>
</tr>
<tr>
<td>1105</td>
<td>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.</td>
</tr>
<tr>
<td>1106</td>
<td>The query processing component received an unknown response while waiting for search results.</td>
</tr>
<tr>
<td>1107</td>
<td>A communication error occurred while the query processing component was attempting to send the query to a query matching node.</td>
</tr>
<tr>
<td>1108</td>
<td>An error occurred while the query processing component was requesting a document summary.</td>
</tr>
<tr>
<td>1109</td>
<td>The retrieval of document summaries took longer than the maximum amount of time permitted for the operation.</td>
</tr>
<tr>
<td>1110</td>
<td>An error occurred while the query processing component was attempting to establish a connection to retrieve document summaries.</td>
</tr>
<tr>
<td>1111</td>
<td>The query processing component received an invalid response while fetching document summaries.</td>
</tr>
<tr>
<td>1112</td>
<td>The query processing component failed to store search result information.</td>
</tr>
<tr>
<td>1113</td>
<td>The query processing component failed to allocate memory for the query.</td>
</tr>
<tr>
<td>1114</td>
<td>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.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>1201</td>
<td>An error occurred while the query processing component was parsing the FQL expression in the query.</td>
</tr>
<tr>
<td>1202</td>
<td>An error occurred during the processing of search results.</td>
</tr>
<tr>
<td>1999</td>
<td>An unspecified error occurred in the query processing component.</td>
</tr>
</tbody>
</table>

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:
4.1.11 Field Collapsing Example

The following example collapses all similar items based on size:

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

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=%22Adventure%20Works%22%20%22Graphic%20Design%20Institute%22%20%22Southridge%20Video%22&collapseon=batvsize

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:

```plaintext
####
#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:

```plaintext
####
#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 R ATI 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 12~12~11~11~10~10~10~10~10~10~9~9~8~8~8~8
####
#NAV NAME float
#NAV DNAM float
#NAV TYPE Float
#NAV UNIT
```
4.2.3 Item Section Example

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

```plaintext
#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:
and(filter(or(docacl:all, docacl:unknown), annotation_class="invisible", annotation_class="sam"), "frisky")
#tl
d#path /cat.txt
d#crawltime
d#processingtime 2009-09-10T15:01:07Z
d#docdatetime 2009-09-10T15:01:11Z
d#size 53
d#docvector
d#documentsignature 166509901429199529
d#hwboost
d#docrank 0
d#siterank 0
d#urldeptrank 630
d#docacl unknown
d#docaclsystemid unknown
da#uthor
d#createdby
d#fileextension TXT
d#isdocument
d#modifiedby
da#account
d#assignto
d#doccomments
d#dockeywords

d#spdocid
d#docsubject
d#created
d#lastmodifiedtime
d#notes
d#siteid

d#sitename
d#sitetitle
da#psiteurl

da#status
d#crawledpropertynames
d#detectedlanguage en
da#companies

da#locations

da#personnames

da#concepts

da#taxonomy
da#companyteaser
da#locationteaser
da#personnameteaser

da#owsmetadatafacetinfo

d#xml

###/

#### 2

d#rank 762
d#ranklog
d#fcoid 0
d#fcoccount 0
d#morehits 0
d#internalid b83fb55f67e2d5f21cfc81ffe6bb5359_sp
da#contentid http://dummy.com/dog.txt
da#contentids http://dummy.com/dog.txt
da#collection sp
da#title dog.txt
da#body  Dogs are often <b>frisky</b>, and usually smell funny.
da#teaser  Dogs are often frisky, and usually smell funny.
4.2.5 No Items Example

The following example shows a query that searches for the word *monkey*:
Here is the search result of the preceding query, which matches no items:

```xml
#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 MSG <KeywordData />
#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:
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:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:types>
            elementFormDefault="qualified"
            <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
                message="tns:IProxyRemote_ProcessRequest_InputMessage" />
            <wsdl:output
                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
            style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
        </soap:operation>
    </wsdl:binding>
</wsdl:definitions>
```
<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

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

A
Abstract data model 11
server 11
Applicability 9
Attribute groups 10
Attributes 10
Automatic resubmission example 33
Automatic resubmission parameter 18

C
Capability negotiation 9
Change tracking 46
Complex types 10

D
Data model - abstract 11
server 11
Dynamic duplicate removal example 34
Dynamic duplicate removal parameters 20

E
Elements
ProcessRequest 12
ProcessRequestResponse 21
Error example 40
Events
local 31
local - server 31
timer 31
timer - server 31
Examples
Automatic resubmission 33
Dynamic duplicate removal 34
Error 40
Field collapsing 34
Filter 32
Freshness boost 32
Item section 36
Limiting results 34
Multiple items 36
No items 39
Query 32
Query language 34
Query refinement 33
Query transform 35
Query type 32
Refinement 35
Similarity 32
Skipping results 34
Sorting 33
Spell-checking 33
Stemming 33
White space preservation 34

Field collapsing example 34
Field collapsing parameter 20
Fields - vendor-extensible 9
Filter example 32
Filter parameter 13
Freshness boost example 32
Freshness boost parameter 14
Full WSDL 43

G
Glossary 6
Groups 10

I
Implementer - security considerations 42
Index of security parameters 42
Informative references 8
Initialization 11
server 11
Interfaces
Query and Result Protocol 11
IProxyRemote_ProcessRequest_InputMessage message 12
IProxyRemote_ProcessRequest_OutputMessage message 12
Item section example 36
Limiting results example 34
Local events 31
server 31

M
Message processing 11
server 11
Messages
attribute groups 10
attributes 10
complex types 10
elements 10
enumerated 10
groups 10
IProxyRemote_ProcessRequest_InputMessage 12
Messages
IProxyRemote_ProcessRequest_OutputMessage 12
namespaces 10
simple types 10
syntax 10
transport 10
Multiple items example 36

N