# [MS-XWDSEARCH]: Web Distributed Authoring and Versioning (WebDAV) Extensions for Search

## Revision Summary

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 12/03/2008 | 1.0 |  | Initial Release. |
| 03/04/2009 | 1.01 |  | Revised and edited technical content. |
| 04/10/2009 | 2.0 |  | Deprecated for Exchange 2010. |
| 07/15/2009 | 3.0 | Major | Changes made for template compliance. |
| 11/04/2009 | 3.1.0 | Minor | Updated the technical content. |
| 02/10/2010 | 3.2.0 | Minor | Updated the technical content. |
| 05/05/2010 | 3.3.0 | Minor | Updated the technical content. |
| 08/04/2010 | 3.3.0 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 11/03/2010 | 3.3.0 | No change | No changes to the meaning, language, or formatting of the technical content. |

# Contents

# 1 Introduction

This protocol specifies extensions to the Web Distributed Authoring and Versioning (WebDAV) protocol [RFC4918] and Web Distributed Authoring and Versioning (WebDAV) Protocol: Server Extensions [MS-WDVSE] to allow clients to request ranges for server-side searches of content. This protocol also allows clients to create persistent **search folders** on the server.

## 1.1 Glossary

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

**Augmented Backus-Naur Form (ABNF)**
**folder**
**soft delete**
**search folder**
**Uniform Resource Locator (URL)**

The following terms are specific to this document:

**query grammar:** A set of definitions of XML elements, attributes, and constraints on their relations and values that defines a set of queries and the intended semantics.

**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.

[MS-OXCDATA] Microsoft Corporation, "Data Structures", April 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List", April 2008.

[MS-WDVSE] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Protocol: Server Extensions", September 2007, http://msdn.microsoft.com/en-us/library/cc250200.aspx

[MS-XWDEXT] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Core Extensions", April 2009.

[MS-XWDFOLD] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Extensions for Folders Support", December 2008.

[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

[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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

[RFC4234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, http://www.ietf.org/rfc/rfc4234.txt

[RFC4918] Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007, http://www.webdav.org/specs/rfc4918.html

[RFC5323] Reschke, J., Ed., Reddy, S., Davis, J., and Babich, A., "Web Distributed Authoring and Versioning (WebDAV) SEARCH", RFC 5323, November 2008, ftp://ftp.rfc-editor.org/in-notes/rfc5323.txt

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, 16 August 2006, edited in place 29 September 2006, http://www.w3.org/TR/2006/REC-xml-20060816/

### 1.2.2 Informative References

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

## 1.3 Overview

This protocol allows clients to request that the server perform a search on the contents of the server and return a specific result set. This protocol also allows the client to create a search folder on the server.

This protocol specifies the following extensions to [RFC4918], [MS-WDVSE], and [RFC2616]:

- An extension to the **Accept-Ranges** header to indicate support for the extensions specified in this document.
- Extensions to the HTTP **Range** and **Content-Range** headers [RFC2616] to allow clients to request a range of rows.
- An XML element in the DAV: namespace that allows the server to indicate the range of rows that were returned.
- A property that can be set on a search folder to indicate the search that is to be performed.
- A new value for PidNameDavResourceType to indicate that a **folder** is a search folder.
- A **query grammar** that is used to express search requests.

## 1.4 Relationship to Other Protocols

This protocol relies on [MS-XWDEXT], [RFC4918], [MS-WDVSE], and [RFC2616]. It also relies on [RFC2818] for data protection services.

All properties are listed in [MS-OXPROPS] and are formatted as specified in [MS-OXCDATA] for use with WebDAV.

## 1.5 Prerequisites/Preconditions

This specification requires a WebDAV server as specified in [RFC4918]. This specification also requires that WebDAV clients have a **URL** that points to the WebDAV server.

## 1.6 Applicability Statement

This protocol can be used to control what results are returned from **SEARCH** or **PROPFIND** requests, and can also be used to create and modify persistent searches in the form of search folders.

## 1.7 Versioning and Capability Negotiation

Clients can determine whether a server supports the extensions to the **Range** and **Content-Range** headers specified in this protocol by sending an OPTIONS command as specified in [RFC2616] to the server and examining the response. If the server supports this protocol, it MUST return an **Accept-Ranges** header set to "rows" in the OPTIONS response, as specified in [RFC2616].

Clients can determine whether a server supports the query grammar specified in this protocol by sending an OPTIONS command to the server and examining the response. If the server supports this protocol, then it MUST return a **DASL** header ([RFC5323] section 3.2) set to "<DAV:sql>" in the OPTIONS response.

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.

# 2 Messages

## 2.1 Transport

This protocol specifies extensions to, and uses the same transport mechanisms as, [RFC4918] and [MS-WDVSE].

## 2.2 Message Syntax

### 2.2.1 Headers

The **Augmented Backus-Naur Form (ABNF)** notation [RFC4234] is used to specify the format of the following headers.

#### 2.2.1.1 Accept-Ranges

The **Accept-Ranges** header is specified in [RFC2616]. This protocol extends the **Accept-Ranges** header by defining a new **range unit**.

```
row-unit = "rows"
```

#### 2.2.1.2 Range

The **Range** header is optional in a **SEARCH** request. This protocol extends the [RFC2616] ABNF for the **Range** header by defining a new row-ranges-specifier. The byte-ranges-specifier rule is specified in [RFC2616].

```
ranges-specifier = byte-ranges-specifier
                   / row-ranges-specifier

row-ranges-specifier = row-unit "=" [row-range-list]

row-range-list = row-range [","  row-range-list]

row-range = row-range-spec / suffix-row-range-spec

row-range-spec = first-row-num "-" [last-row-num]

first-row-num  = 1*DIGIT
last-row-num   = 1*DIGIT
suffix-row-range-spec = "-" suffix-length
suffix-length = 1*DIGIT
row-unit = "rows"
```

Examples:

```
Range: rows=20-39
Range: rows=0-5, 10-15, 20-25, 30-35
Range: rows=-50   (the last 50 rows)
Range: rows=228-  (all rows from the 228th to the end of the result set)
Range: rows=0-4,-5  (the first and last five rows)
```

### 2.2.1.3 Content-Range

The **Content-Range** header is specified in [RFC2616] and extended here. The byte-content-range-spec rule is specified in [RFC2616].

```
Content-Range = "Content-Range" ":" content-range-spec

content-range-spec      = byte-content-range-spec / row-content-range-spec

row-content-range-spec = row-unit row-range-list [";" "total=" total-rows] / "total=" total-
rows

row-range-list = row-range ["," row-range-list]

row-range = row-range-spec / suffix-row-range-spec

row-range-spec = first-row-num "-" [last-row-num]

suffix-row-range-spec = "-" suffix-length

first-row-num  = 1*DIGIT
last-row-num   = 1*DIGIT
suffix-length = 1*DIGIT
total-rows = 1*DIGIT / "*"
row-unit = "rows"
```

Examples:

```
Content-Range: rows 0-5; total=200
Content-Range: rows 0-5, 7-10; total=*
Content-Range: rows 0-50; total=51
Content-Range: total=0
```

## 2.2.2 XML Element

### 2.2.2.1 DAV:contentrange

The DAV:contentrange XML element is similar in format and use to the **Content-Range** header.

```
DAV:contentrange = "<DAV:contentrange>" content-range-spec "</DAV:contentrange>"

content-range-spec = first-row-num "-" last-row-num

first-row-num = 1*DIGIT
last-row-num = 1*DIGIT
```

## 2.2.3 Properties

The following subsections provide the full property descriptions from [MS-OXPROPS]. However, the property names listed as "Alternate names" are the names used in WebDAV commands.

#### 2.2.3.1 PidNameDavSearchRequest

**DAV Property name:** DAV:searchrequest

**Data type: PtypString**

This property indicates the search that the search folder will perform. The syntax of PidNameDavSearchRequest is specified in section 2.2.4. This property MUST be present on search folders and MUST NOT be present on other types of resources. For additional information, see [MS-OXPROPS] section 2.448.

#### 2.2.3.2 PidNameDavSearchType

**DAV Property name:** DAV:searchtype

**Data type: PtypString**

This property indicates what type of search the search folder is performing. The value MUST be either "dynamic" or "static". The property is set by the server and is read-only to the client. This property MUST be present on search folders and MUST NOT be present on other types of resources. For additional information, see [MS-OXPROPS] section 2.449.

##### 2.2.3.2.1 Dynamic

The server MUST set PidNameDavSearchType to "dynamic" when the server updates the contents of the search folder as content changes on the server.

##### 2.2.3.2.2 Static

The server MUST set PidNameDavSearchType to "static" when the server does not update the contents of the search folder as content changes on the server.

#### 2.2.3.3 PidNameDavResourceType

This protocol specifies a new value for PidNameDavResourceType. This value is "<DAV:searchresults/>". If this value and "<DAV:collection/>" is present in the PidNameDavResourceType property on a folder, then that folder is a search folder. PidNameDavResourceType is specified in [RFC4918] as <DAV:resourcetype>.

### 2.2.4 Query Grammar

The grammar required for PidNameDavSearchRequest is defined by the following ABNF notation.

```
search-request = select-clause
                 from-predicate
                 [where-clause]
                 [( order-by-clause [group-by-predicate]  ) /
                  (group-by-predicate [order-by-clause])]

select-clause = "SELECT" select-items
select-items = ["ALL" / "DISTINCT"] ( "*" / select-specific )
select-specific = property-expression *["," property-expression]
property-expression = property-name / property-name " AS " alias
property-name = DQUOTE 1*VCHAR DQUOTE
alias = DQUOTE 1*VCHAR DQUOTE
```

```
from-predicate = "FROM" from-clause
from-clause = scope-specification / target-url

scope-specification = "SCOPE" "(" ["'" scope-arguments "'"] *["," "'" scope-arguments "'"]
")" ["AS" view-spec]
scope-arguments = [traversal-depth] target-url
traversal-depth = (("SHALLOW" / "DEEP" / "HIERARCHICAL" / "SOFTDELETED") "TRAVERSAL OF")
target-url = (DQUOTE 1*VCHAR DQUOTE) / ("'" 1*VCHAR "'")
view-spec = DQUOTE 1*VCHAR DQUOTE

where-clause = "WHERE" condition-expression
condition-expression = boolean-expression / ( condition-expression "OR" boolean-expression )
boolean-expression = boolean-factor / ( boolean-expression "AND" boolean-factor )
boolean-factor = ["NOT"] boolean-item [test]
test = "IS" ["NOT"] evaluation
evaluation = "TRUE" / "FALSE"
boolean-item = predicate / ( "(" condition-expression ")" )
predicate = comparison-predicate / null-predicate / like-predicate

null-predicate = property-name "IS" ["NOT"] "NULL"

like-predicate = propertyname-expression ["NOT"] "LIKE" wildcard-search-pattern
wildcard-search-pattern = like-prefix / like-contains
like-prefix = "'" 1*VCHAR "%'"
like-contains = "'%" 1*VCHAR "%'"

comparison-predicate = propertyname-expression operator value-expression
propertyname-expression = property-name / property-cast
property-cast = "CAST" "(" property-name "AS" data-type ")"
value-expression = property-value / value-cast
value-cast = "CAST" "(" property-value "AS" data-type ")"
operator = "=" / "<>" / "!=" / "<" / ">" / "<=" / ">="

order-by-clause = "ORDER BY" order-clause
order-clause = sort-specification *["," sort-specification]
sort-specification = property-name [sort-order]
sort-order = "ASC" / "DESC"

group-by-predicate = "GROUP BY" group-clause
group-clause = group-spec *["," group-spec]
group-spec = property-name

property-value = evaluation / 1*DIGIT / ("'" 1*VCHAR "'")

data-type = ( [WebDAV-type-modifier] WebDAV-Type ) / DBType

WebDAV-type-modifier = "limited." /
                       "mv."

WebDAV-Type =  "boolean" /
               "bin.hex" /
               "bin.base64" /
               "char" /
               "dateTime" /
               "dateTime.tz" /
               "dateTime.rfc1123" /
               "fixed.14.4" /
               "float" /
               "i1" /
```

```
                    "i2" /
                    "i4" /
                    "i8" /
                    "int" /
                    "number" /
                    "r4" /
                    "r8" /
                    "string" /
                    "ui1" /
                    "ui2" /
                    "ui4" /
                    "ui8" /
                    "uri" /
                    "uuid"

DBType =  "DBTYPE_BOOL" /
          "DBTYPE_BSTR" /
          "DBTYPE_BYTES" /
          "DBTYPE_CY" /
          "DBTYPE_DATE" /
          "DBTYPE_FILETIME" /
          "DBTYPE_GUID" /
          "DBTYPE_I1" /
          "DBTYPE_I2" /
          "DBTYPE_I4" /
          "DBTYPE_I8" /
          "DBTYPE_R4" /
          "DBTYPE_R8" /
          "DBTYPE_STR" /
          "DBTYPE_UI1" /
          "DBTYPE_UI2" /
          "DBTYPE_UI4" /
          "DBTYPE_UI8" /
          "DBTYPE_WSTR"
```

### 2.2.4.1 select-clause

The select-clause specifies what properties are to be returned in a search. It MUST be present in a search request.

#### 2.2.4.1.1 Syntax

```
SELECT [ALL | DISTINCT] list
```

The server MUST ignore the ALL and DISTINCT prefixes.

##### 2.2.4.1.1.1 list

The list parameter can have the following values:

| Value | Description |
|---|---|
| * | The server MUST return the properties defined in the PidNameExchDataExpectedContentClass ([MS-OXPROPS] section 2.467) property on the folder being searched. |

| Value | Description |
|---|---|
| A comma-delimited list of WebDAV properties. | The server MUST return the requested properties in the result set. |

#### 2.2.4.2 from-predicate

The from-predicate specifies the folders on which the search will be performed. It MUST be present in a search request.

##### 2.2.4.2.1 Syntax

```
FROM scope | url
```

###### 2.2.4.2.1.1 scope

The *scope* parameter can be used to control the location and depth of a search. If the *scope* parameter is used, the *url* parameter MUST NOT be used. The syntax for the *scope* parameter is specified in section 2.2.4.2.2.

###### 2.2.4.2.1.2 url

The *url* parameter can be used to control the location of a search. The search that will be performed will be a SHALLOW search as specified in section 2.2.4.2.2.1.1. The *url* parameter is a **Uniform Resource Locator (URL)** of the folder to be searched. If this parameter is omitted, then the Request-URI [RFC2616] is used in its place.

##### 2.2.4.2.2 scope-specification

The scope-specification element is an optional part of the from-predicate and is used to specify the location and depth of a search.

###### 2.2.4.2.2.1 Syntax

```
SCOPE('[depth TRAVERSAL OF] "url"') [AS view]
```

###### 2.2.4.2.2.1.1 depth

The *depth* parameter specifies the depth of the search. It can be one of the values in the following table:

| Value | Description |
|---|---|
| SHALLOW | The requested search will be performed on items contained in the folder being searched, but not on items contained in subfolders within that folder. |
| DEEP | The requested search will be performed on items contained in the folder being searched and on items contained in subfolders, all the way to the bottom of the folder hierarchy.<1> |
| HIERARCHICAL | The requested search will be performed on folder items in the folder being searched. |

| Value | Description |
|---|---|
| SOFTDELETED | The requested search will be performed on the **soft deleted** contents of the folder. |

If no depth is specified, the server MUST treat the request as though DEEP TRAVERSAL OF was specified.

###### 2.2.4.2.2.1.2 url

The *url* parameter can be used to control the location of a search. The *url* parameter is a **Uniform Resource Locator (URL)** of the folder to be searched. If this parameter is omitted, then the Request-URI [RFC2616] is used in its place.

###### 2.2.4.2.2.1.3 view

The server MUST ignore the *view* parameter.

### 2.2.4.3 where-clause

The where-clause specifies search criteria for the search request. It is optional in a search request. If the where-clause is included in a search request, the server MUST only return results that match the search criteria.

#### 2.2.4.3.1 Syntax

```
WHERE expression AND | OR expression
```

##### 2.2.4.3.1.1 expression

The *expression* parameter can take the form of any of the following predicates: comparsion-predicate, null-predicate, or like-predicate.

#### 2.2.4.3.2 comparison-predicate

The comparison-predicate is used to do a simple comparison of values.

##### 2.2.4.3.2.1 Syntax

property-name-expression operator value-expression

###### 2.2.4.3.2.1.1 property-name-expression

The *property-name-expression* parameter can take two forms. It can either be a property name, or it can be a property-cast function using the *property* parameter.

###### 2.2.4.3.2.1.2 operator

The *operator* parameter can be one of the values from the following table:

| Operator | Description |
|---|---|
| = | The item on the left of the operator is tested for equality with the item on the right of the operator. If the items are equal, the criteria is satisfied. |

| Operator | Description |
|---|---|
| <> | The item on the left of the operator is tested for equality with the item on the right of the operator. If the items are not equal, the criteria is satisfied.<br>**Note** When this operator is used in XML, the '<' is encoded as '&lt;' as specified in [XML]. |
| != | The item on the left of the operator is tested for equality with the item on the right of the operator. If the items are not equal, the criteria is satisfied. |
| < | The relative value of the item on the left of the operator is compared to the relative value of the item on the right of the operator. If the value of the item on the left is less than the value of the item on the right, the criteria is satisfied.<br>**Note** When this operator is used in XML, the '<' is encoded as '&lt;' as specified in [XML]. |
| > | The relative value of the item on the left of the operator is compared to the relative value of the item on the right of the operator. If the value of the item on the left is greater than the value of the item on the right, the criteria is satisfied. |
| <= | The relative value of the item on the left of the operator is compared to the relative value of the item on the right of the operator. If the value of the item on the left is less than or equal to the value of the item on the right, the criteria is satisfied.<br>**Note** When this operator is used in XML, the '<' is encoded as '&lt;' as specified in [XML]. |
| >= | The relative value of the item on the left of the operator is compared to the relative value of the item on the right of the operator. If the value of the item on the left is greater than or equal to the value of the item on the right, the criteria is satisfied. |

### 2.2.4.3.2.1.3 value-expression

The *value-expression* parameter can take two forms. It can either be a literal value, or it can be a property-cast function using the *value* parameter.

### 2.2.4.3.2.1.4 property-cast

A property-cast function is used to specify a specific datatype to be used to interpret a value.

### 2.2.4.3.2.1.4.1 Syntax

```
CAST("property | value" AS "datatype")
```

### 2.2.4.3.2.1.4.1.1 property

The *property* parameter is used to specify a property name. When this parameter is used, the value of that property will be cast to the datatype specified in the *datatype* parameter.

### 2.2.4.3.2.1.4.1.2 value

The *value* parameter is used to specify a literal value. When this parameter is used, the literal value will be cast to the datatype specified in the *datatype* parameter.

##### 2.2.4.3.2.1.4.1.3 datatype

The *datatype* parameter can be any of the "Alternate Names" for the Property Value Types specified in [MS-OXCDATA] section 2.12.1, or any of the WebDAV Property Value Types specified in [MS-OXCDATA] section 2.11.1.5. See the "data-type" ABNF rule in section 2.2.4

#### 2.2.4.3.3 null-predicate

The null-predicate specifies that a property's value is to be tested to determine if it is null or, alternatively, not null.

##### 2.2.4.3.3.1 Syntax

propertyname IS [NOT] NULL

##### 2.2.4.3.3.1.1 propertyname

The *propertyname* parameter specifies which property's value is to be tested. This parameter is a WebDAV property name.

#### 2.2.4.3.4 like-predicate

The like-predicate specifies a string to check for within a specific property. The search does a character-by-character comparison.

##### 2.2.4.3.4.1 Syntax

```
propertyname LIKE wildcard-value
```

##### 2.2.4.3.4.1.1 propertyname

The *propertyname* parameter specifies which property value is to be included in the search. It is a WebDAV property name surrounded by double-quotes.

##### 2.2.4.3.4.1.2 wildcard-value

The *wildcard-value* parameter specifies the character pattern to be searched for. It MUST be surrounded by single quotes and the last character in the pattern MUST be a percent-sign ('%').

### 2.2.4.4 order-by-clause

The order-by-clause specifies what properties are to be used to sort the search results. It is optional in search requests.

#### 2.2.4.4.1 Syntax

```
ORDER BY propertyname sortorder [, propertyname sortorder]
```

##### 2.2.4.4.1.1 propertyname

The *propertyname* parameter specifies which property value is to be used to sort the results. It is a WebDAV property name surrounded by double-quotes.

#### 2.2.4.4.1.2 Sortorder

The *sortorder* parameter specifies the order the results are sorted in. It MUST be one of the following values:

| Value | Description |
|---|---|
| ASC | The results are sorted in ascending order. For string properties this is A-Z, and for numerical properties this is lesser values - greater values. |
| DESC | The results are sorted in descending order. For string properties this is Z-A, and for numerical properties this is greater values - lesser values. |

### 2.2.4.5 group-by-predicate

The group-by-predicate specifies what properties are to be used to group the search results. It is optional in search requests.

#### 2.2.4.5.1 Syntax

```
GROUP BY propertyname [, propertyname]
```

#### 2.2.4.5.1.1 propertyname

The *propertyname* parameter specifies which property value is to be used to group the results. It is a WebDAV property name surrounded by double-quotes.

# 3 Protocol Details

## 3.1 Client Details

### 3.1.1 Abstract Data Model

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

### 3.1.2 Timers

None.

### 3.1.3 Initialization

None.

### 3.1.4 Higher-Layer Triggered Events

None.

### 3.1.5 Message Processing Events and Sequencing Rules

#### 3.1.5.1 Accept-Ranges

The client can send an OPTIONS request to the server and examine the response for the **Accept-Ranges** header. The client MUST NOT use the extensions to the **Range** header specified in section 2.2.1.2 unless the **Accept-Ranges** header contains the value "rows".

#### 3.1.5.2 Range

The client can send an optional **Range** header in a SEARCH or PROPFIND request to limit the results that are returned by the server.

#### 3.1.5.3 Content-Range

The client can examine server responses to SEARCH and PROPFIND for the **Content-Range** header to determine what rows were returned. This information can alternatively be found in the **DAV:contentrange** element in the XML body of the response.

#### 3.1.5.4 DAV:contentrange

The client can examine the XML body returned in the server response to SEARCH or PROPFIND for the **DAV:contentrange** element to determine what rows were returned. This information can alternatively be found in the **Content-Range** header of the response.

#### 3.1.5.5 PidNameDavSearchRequest

The client can send a MKCOL command to create a search folder. The request MUST include an XML body with a **propertyupdate** element, as specified in [RFC4918], for the **PROPPATCH** method.

The **propertyupdate** element MUST set the PidNameDavSearchRequest property with the desired search.

The client can update the PidNameDavSearchRequest property to modify the search that a search folder will perform.

The client can query the PidNameDavSearchRequest property to determine what search a search folder is performing.

#### 3.1.5.6 PidNameDavSearchType

The client can query the PidNameDavSearchType property on a search folder to determine whether the server will update the contents of the search folder updates dynamically.

#### 3.1.5.7 PidNameDavResourceType

The client can query the **DAV:resourcetype** property on a folder to determine whether the folder is a search folder. The client can check for the values "<DAV:collection/>" and "<DAV:searchresults/>", which indicate that the folder is a search folder.

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 Server Details

### 3.2.1 Abstract Data Model

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

### 3.2.2 Timers

None.

### 3.2.3 Initialization

None.

### 3.2.4 Higher-Layer Triggered Events

None.

## 3.2.5 Message Processing Events and Sequencing Rules

### 3.2.5.1 Accept-Ranges

If the server supports the extensions to the **Range** and **Content-Range** headers specified in this protocol, then it MUST return an **Accept-Ranges** header set to "rows" in response to an OPTIONS request.

### 3.2.5.2 Range

If the server receives a SEARCH or PROPFIND request with a **row-ranges-specifier**, then it MUST limit the results returned in the response accordingly. The server MUST add a **DAV:contentrange** element to the XML response indicating the rows that were returned, and it MUST return a **Content-Ranges** header, as specified in section 3.2.5.3.

If the contents of the **Range** header are incorrectly formed, then the server MUST return an HTTP status 400 "Bad Request" error.

### 3.2.5.3 Content-Range

The server MUST return a **Content-Range** header in response to a SEARCH or PROPFIND request if the request contained a **Range** header.

### 3.2.5.4 DAV:contentrange

The server MUST add a **DAV:contentrange** XML element to the XML body in the response to a SEARCH or PROPFIND request when the request contained a **Range** header.

### 3.2.5.5 PidNameDavSearchRequest

If the server receives a request to create a folder, as specified in [MS-XWDFOLD], that includes a request to set PidNameDavSearchRequest, then it MUST create a search folder to execute the requested search, or return an error indicating the reason it cannot fulfill the request.

PidNameDavSearchRequest is a read-only property. If the server receives a request to update the PidNameDavSearchRequest property on a search folder, then it MUST return a 403 Forbidden error.

### 3.2.5.6 PidNameDavSearchType

The server MUST set PidNameDavSearchType to indicate whether it will update the contents of the search folder dynamically as content changes on the server.

The server MUST NOT allow clients to set this property, and SHOULD return an HTTP 403 "Forbidden" error within an HTTP 207 "Multi-Status" response.

### 3.2.5.7 PidNameDavResourceType

The server MUST include the values "<DAV:searchresults/>" and "<DAV:collection/>" in the PidNameDavResourceType property on search folders.

## 3.2.6 Timer Events

None.

### 3.2.7 Other Local Events

None.

# 4 Protocol Examples

## 4.1 Requesting a Range of Rows in a SEARCH Request

Alex Robinson is using a WebDAV client to search his Inbox. He wants to limit the search results to the third and fourth items only.

The client begins by creating a SEARCH request, as specified in [MS-WDVSE]. Then, the client adds the **Range** header with a value of "rows=2-3" to limit the results to the desired subset.

```
SEARCH /exchange/alex/inbox HTTP/1.1
Range: rows=2-3
Content-Type: text/xml

<?xml version="1.0"?>
<g:searchrequest xmlns:g="DAV:">
  <g:sql>
SELECT "DAV:displayname" FROM SCOPE('SHALLOW TRAVERSAL OF "/exchange/alex/inbox"')
  </g:sql>
</g:searchrequest>
```

The server processes the SEARCH request and returns the results. In this case, there were nine total items in Alex's Inbox. The server adds the **Content-Range** header with a value of "rows 2-3; total=9," indicating that row numbers 2 and 3 were returned, and that there are nine total rows that could have been returned. The server also adds the DAV:contentrange element to the XML body of the response, with a value of "2-3."

```
HTTP/1.1 207 Multi-Status
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: text/xml
Content-Range: rows 2-3; total=9
Accept-Ranges: rows
Date: Tue, 07 Oct 2008 04:22:48 GMT

<?xml version="1.0"?>
<a:multistatus xmlns:a="DAV:">
<a:contentrange>2-3</a:contentrange>
<a:response><a:href>http://server/exchange/alex/Inbox/Test3.EML
</a:href><a:propstat><a:status>HTTP/1.1 200 OK</a:status>
<a:prop><a:displayname>Test3.EML</a:displayname></a:prop></a:propstat></a:response>
<a:response>
<a:href> http://server/exchange/alex/Inbox/Test4.EML </a:href><a:propstat><a:status>HTTP/1.1
200 OK</a:status>
<a:prop><a:displayname> Test4.EML</a:displayname></a:prop></a:propstat>
</a:response>
</a:multistatus>
```

## 4.2 Creating a Search Folder

Alex Robinson wants to create a persistent search folder inside his Inbox that will find all messages with "ACTION REQUIRED" in the subject. The client forms a MKCOL request and sets PidNameDavSearchRequest to an appropriate value.

```
MKCOL /exchange/alex/inbox/Action HTTP/1.1
Content-type: text/xml

<?xml version="1.0"?>
<a:propertyupdate xmlns:a="DAV:">
  <a:set>
  <a:prop>
  <a:searchrequest>
    <a:sql>
      SELECT * FROM scope('shallow traversal of "http://server/exchange/alex/inbox"')
      WHERE "urn:schemas:httpmail:subject" LIKE '%ACTION REQUIRED%'
    </a:sql>
  </a:searchrequest>
  </a:prop>
  </a:set>
</a:propertyupdate>
```

The server processes the request and successfully creates the search folder.

```
HTTP/1.1 207 Multi-Status
Date: Thu, 16 Oct 2008 16:04:10 GMT
Content-Type: text/xml
Content-Length: 293

<?xml version="1.0" ?>
<a:multistatus xmlns:a="DAV:">
  <a:response>
    <a:href>http://server/exchange/alex/Inbox/Action</a:href>
    <a:status>HTTP/1.1 201 Created</a:status>
    <a:propstat>
      <a:status>HTTP/1.1 200 OK</a:status>
      <a:prop>
        <a:searchrequest />
      </a:prop>
    </a:propstat>
  </a:response>
</a:multistatus>
```

## 4.3 Verifying Search Type

The client then checks to see whether the contents of the newly-created search folder will be updated dynamically. The client creates a PROPFIND request to check the value of PidNameDavSearchType.

```
PROPFIND /exchange/alex/inbox/Action HTTP/1.1
Content-type: text/xml
Translate: f
Depth: 0

<?xml version="1.0"?>
<a:propfind xmlns:a="DAV:">
  <a:prop><a:searchtype/></a:prop>
</a:propfind>
```

The server is dynamically updating the contents of the search folder as new content arrives. It returns a response indicating that the search type is dynamic.

```
HTTP/1.1 207 Multi-Status
Date: Thu, 16 Oct 2008 16:12:45 GMT
Content-Type: text/xml

<?xml version="1.0" ?>
<a:multistatus xmlns:b="urn:uuid:c2f41010-65b3-11d1-a29f-00aa00c14882/" xmlns:c="xml:"
xmlns:a="DAV:">
  <a:response>
    <a:href>http://server/exchange/alex/Inbox/Action/</a:href>
    <a:propstat>
      <a:status>HTTP/1.1 200 OK</a:status>
      <a:prop>
        <a:searchtype>dynamic</a:searchtype>
      </a:prop>
    </a:propstat>
  </a:response>
</a:multistatus>
```

# 5 Security

## 5.1 Security Considerations for Implementers

There are no special security concerns specific to the WebDAV Extensions for Search Protocol. General security considerations pertaining to the underlying transport apply, as specified in [RFC4918] and [MS-WDVSE].

## 5.2 Index of Security Parameters

None.

# 6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Exchange Server 2003
- Microsoft® Exchange Server 2007

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. 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 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 product does not follow the prescription.

<1> Section 2.2.4.2.2.1.1: Exchange 2003 and Exchange 2007 do not support DEEP traversals in the public folder store.

# 7 Change Tracking

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

# 8 Index