

[MS-SPSCRWL]:

SPSCrawl Web Service Protocol

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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
4/25/2008	0.2	Editorial	Revised and edited the technical content
6/27/2008	1.0	Major	Revised and edited the technical content
10/6/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Major	Updated and revised the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Minor	Clarified the meaning of the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	Major	Significantly changed the technical content.
3/18/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.05	None	No changes to the meaning, language, or formatting of the technical content.
4/11/2012	2.05	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.05	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.05	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.6	Minor	Clarified the meaning of the technical content.
2/11/2013	2.6	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/30/2013	2.7	Minor	Clarified the meaning of the technical content.
11/18/2013	2.7	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.7	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.7	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	2.7	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.8	Minor	Clarified the meaning of the technical content.
2/26/2016	3.0	Major	Significantly changed the technical content.

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	9
1.5	Prerequisites/Preconditions	10
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments.....	10
2	Messages.....	11
2.1	Transport	11
2.2	Common Message Syntax	11
2.2.1	Namespaces	11
2.2.2	Messages.....	11
2.2.3	Elements	12
2.2.4	Complex Types.....	12
2.2.4.1	ArrayOf_PortalPropValue.....	12
2.2.4.2	_PortalPropValue	12
2.2.4.3	ArrayOf_PortalItem	13
2.2.4.4	_PortalItem	13
2.2.5	Simple Types	14
2.2.6	Attributes	14
2.2.7	Groups	14
2.2.8	Attribute Groups.....	14
3	Protocol Details.....	15
3.1	Server Details.....	15
3.1.1	Abstract Data Model.....	15
3.1.2	Timers	16
3.1.3	Initialization.....	16
3.1.4	Message Processing Events and Sequencing Rules	16
3.1.4.1	EnumerateBucket.....	16
3.1.4.1.1	Messages	16
3.1.4.1.1.1	EnumerateBucketSoapIn.....	16
3.1.4.1.1.2	EnumerateBucketSoapOut	17
3.1.4.1.2	Elements	17
3.1.4.1.2.1	EnumerateBucket	17
3.1.4.1.2.2	EnumerateBucketResponse	17
3.1.4.2	EnumerateFolder	17
3.1.4.2.1	Messages	18
3.1.4.2.1.1	EnumerateFolderSoapIn	18
3.1.4.2.1.2	EnumerateFolderSoapOut	18
3.1.4.2.2	Elements	18
3.1.4.2.2.1	EnumerateFolder	18
3.1.4.2.2.2	EnumerateFolderResponse	18
3.1.4.3	GetBucket	19
3.1.4.3.1	Messages	19
3.1.4.3.1.1	GetBucketSoapIn	19
3.1.4.3.1.2	GetBucketSoapOut.....	19
3.1.4.3.2	Elements.....	19
3.1.4.3.2.1	GetBucket.....	19

3.1.4.3.2.2	GetBucketResponse	20
3.1.4.4	GetItem	20
3.1.4.4.1	Messages	20
3.1.4.4.1.1	GetItemSoapIn	20
3.1.4.4.1.2	GetItemSoapOut	20
3.1.4.4.2	Elements	21
3.1.4.4.2.1	GetItem	21
3.1.4.4.2.2	GetItemResponse	21
3.1.4.5	GetSite	22
3.1.4.5.1	Messages	22
3.1.4.5.1.1	GetSiteSoapIn	22
3.1.4.5.1.2	GetSiteSoapOut	22
3.1.4.5.2	Elements	22
3.1.4.5.2.1	GetSite	22
3.1.4.5.2.2	GetSiteResponse	23
3.1.4.5.3	Complex Types	23
3.1.4.5.3.1	_PortalSite	23
3.1.5	Timer Events	23
3.1.6	Other Local Events	23
4	Protocol Examples	24
4.1	People Search	24
4.1.1	Data	24
4.1.2	Full Crawl	24
5	Security	31
5.1	Security Considerations for Implementers	31
5.2	Index of Security Parameters	31
6	Appendix A: Full WSDL	32
7	Appendix B: Product Behavior	36
8	Change Tracking	37
9	Index	39

1 Introduction

The SPSCrawl Web Service Protocol allows protocol clients to read the value of item properties for any items on the protocol server.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

bucket: A collection of items that were requested by a search application during a **crawl**. An item can be a person, a document, or any other type of item that can be crawled.

category: A custom string that is used to group one or more documents.

crawl: The process of traversing a URL space to acquire items to record in a search catalog.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

front-end web server: A server that hosts webpages, performs processing tasks, and accepts requests from protocol clients and sends them to the appropriate back-end server for further processing.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [\[RFC4122\]](#) or [\[C706\]](#) must be used for generating the **GUID**. See also universally unique identifier (UUID).

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [\[SSL3\]](#) and [\[RFC5246\]](#).

item: A unit of content that can be indexed and searched by a search application.

language code identifier (LCID): A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

partition: An area within a shared services database, such as an area that isolates different tenants within a service, or the process of creating such an area in a shared services database.

search folder: A collection of related items to be crawled by a search service.

Security Support Provider Interface (SSPI): A Windows-specific API implementation that provides the means for connected applications to call one of several security providers to establish authenticated connections and to exchange data securely over those connections. This is the Windows equivalent of Generic Security Services (GSS)-API, and the two families of APIs are on-the-wire compatible.

service application: A middle-tier application that runs without any user interface components and supports other applications by performing tasks such as retrieving or modifying data in a database.

site: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

SOAP action: The HTTP request header field used to indicate the intent of the SOAP request, using a **URI** value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [\[SOAP1.2-1/2007\]](#) section 5.3 for more information.

SOAP fault: A container for error and status information within a SOAP message. See [\[SOAP1.2-1/2007\]](#) section 5.4 for more information.

SQL authentication: One of two mechanisms for validating attempts to connect to instances of SQL Server. In SQL authentication, users specify a SQL Server login name and password when they connect. The SQL Server instance ensures that the login name and password combination are valid before permitting the connection to succeed.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [\[RFC3986\]](#).

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

user profile store: A database that stores information about each user profile.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [\[RFC3986\]](#). A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [\[XMLNS-2ED\]](#).

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [\[XML\]](#).

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

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.

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

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

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

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

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

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

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

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

1.2.2 Informative References

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

[MS-WSSFO2] Microsoft Corporation, "[Windows SharePoint Services \(WSS\): File Operations Database Communications Version 2 Protocol](#)".

1.3 Protocol Overview (Synopsis)

This protocol allows protocol clients to read the value of any **item** within the context of a **site** or **service application**. The following diagram shows data flow between protocol client and protocol server.

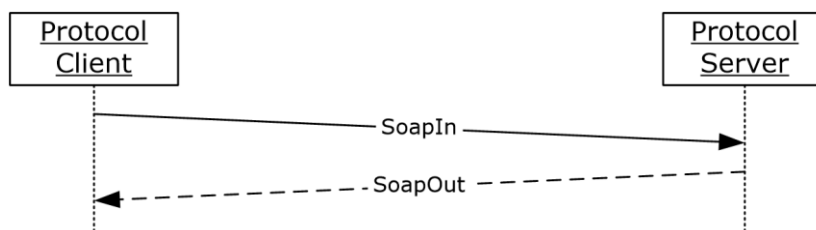


Figure 1: SPS Crawl Web Service Protocol data flow diagram (basic)

Additional details about the protocol are displayed in the following figure.

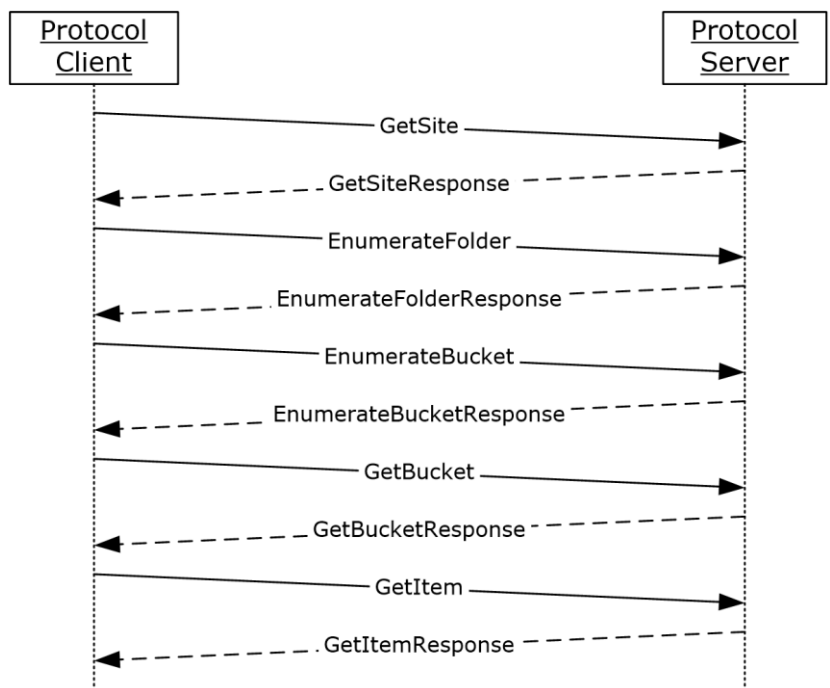


Figure 2: SPS Crawl Web Service Protocol data flow diagram (detailed)

The protocol client requests that the protocol server provide information about the site(2). On receiving this information, the protocol client requests that the protocol server provide a list of all **search folders**. After the protocol server provides information about all the search folders, the protocol client requests that the protocol server enumerate **buckets** in each search folder. Once this information is provided by the protocol server, the protocol client requests that the protocol server provide details about each item within a bucket.

1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using HTTP, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol:

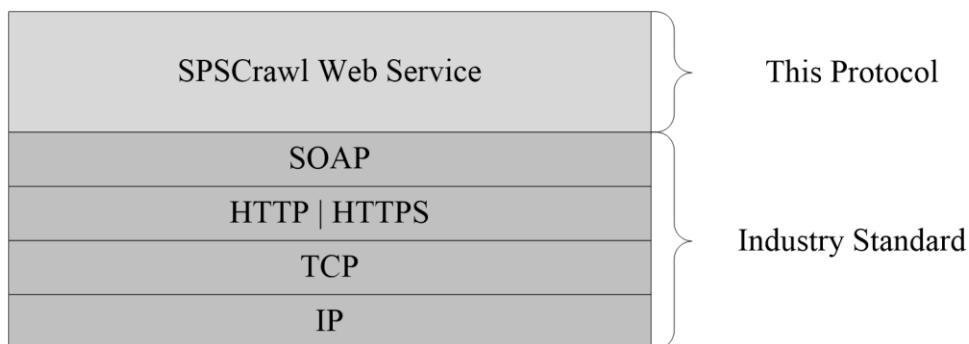


Figure 3: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that is identified by a **URL** that is known by protocol clients. The protocol **endpoint** is formed by appending SPSCrawl.asmx to the URL of the protocol server, for example: `http://www.contoso.com:56737/SharedServices1/_vti_bin/spscrawl.asmx`.

1.6 Applicability Statement

This protocol allows a protocol client to read up to 10 million items.

1.7 Versioning and Capability Negotiation

Versions of the data structures or stored procedures in the database need to be the same as expected by the **front-end Web server**. If the stored procedures do not provide the calling parameters or return values as expected, the results of the call are indeterminate.

The version negotiation process for this protocol is identical to the process described in [\[MS-WSSFO2\]](#) section 1.7.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

2.1 Transport

Protocol servers **MUST** support SOAP over HTTP. Additionally, protocol servers **SHOULD** support SOAP over Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS) for securing communication with clients.

This protocol uses the SOAP messaging protocol for formatting requests and responses as specified in [\[SOAP1.1\]](#) section 4 or in [\[SOAP1.2/1\]](#) section 5. Protocol server faults are returned either using an HTTP status code as specified in [\[RFC2616\]](#), section 10, or using a **SOAP fault** as specified either in [\[SOAP1.1\]](#) section 4.4 or in [\[SOAP1.2/1\]](#) section 5.4.

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses XML Schema as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and WSDL as defined 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. These namespaces are described in the following table.

Prefix	Namespace URI	Reference
mime	http://schemas.xmlsoap.org/wsdl/mime	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soapenc	http://schemas.xmlsoap.org/soap/encoding/	
s0	http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/	
tm	http://microsoft.com/wsdl/mime/textMatching/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

2.2.4 Complex Types

The following table summarizes the set of common XML Schema complex type definitions defined by this specification. XML Schema complex type definitions that are specific to a particular operation are described with the operation.

Complex type	Description
ArrayOf_PortalPropValue	Holds an array of _PortalPropValue elements (section 2.2.4.2).
_PortalPropValue	Holds the value of the property of an item.
ArrayOf_PortalItem	Holds an array of _PortalItem elements (section 2.2.4.4).
_PortalItem	Holds the identifier and last modification time for the item.

2.2.4.1 ArrayOf_PortalPropValue

Holds an array of **_PortalPropValue** (section [2.2.4.2](#)) elements.

```
<s:complexType name="ArrayOf_PortalPropValue">
  <s:sequence>
    <s:element name="_PortalPropValue" type="s0:_PortalPropValue" minOccurs="0"
maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

_PortalPropValue: Individual **_PortalPropValue** (section 2.2.4.2) elements.

If a property is of type **multistring**, the first **multistring** entry MUST contain the first value of the property. The entries following the **multistring** entry MUST each contain one of the remaining values, with their **type** empty, their **URI** empty, and **count** set to zero. The total number of the **multistring** entry and the subsequent entries with their **type** empty and **count** set to zero MUST equal the value of the **count** property of the first **multistring** entry. If a property is of any other type, it MUST contain the value of that property, as defined in **_PortalPropValue** (section 2.2.4.2).

2.2.4.2 _PortalPropValue

Holds property value information.

```
<s:complexType name="_PortalPropValue">
  <s:sequence>
    <s:element name="URI" type="s:string" minOccurs="0"/>
    <s:element name="Value" type="s:string" minOccurs="0"/>
    <s:element name="Type" type="s:string" minOccurs="0"/>
    <s:element name="Count" type="s:int"/>
    <s:element name="UseLCID" type="s:boolean"/>
    <s:element name="LCID" type="s:unsignedInt"/>
  </s:sequence>
</s:complexType>
```

URI: The **URI** of the requested item. The name of the property. This element **MUST** be present.

Value: The value of the property. This element **MUST** be present.

Type: The type of the property, which **MUST** be one of the values listed in the following table.

Value	Description
boolean	The property contains the value 0 or 1 encoded in the current culture of the property as a string.
String	The property is a string.
datetime	The property contains a datetime value encoded in universal datetime pattern, that can be sorted ("yyyy'-MM'-dd HH':'mm':'ss'Z'") as a string.
int32	The property contains a 32-bit integer encoded in the current culture of the property as a string.
int64	The property contains a 64-bit integer encoded in the current culture of the property as a string.
Float	The property contains a single-precision floating point number encoded in the current culture of the property as a string.
multistring	The property contains multiple strings as defined in ArrayOf_PortalPropValue (section 2.2.4.1).

Count: The number of the values returned. **MUST** be 1 if the object type is not **multistring**; otherwise, the **Count** of the first **multistring** entry represents the number of represented values, and count fields of the entries following the first entry are set to zero.

UseLCID: Indicates whether localization information is available in the **language code identifier (LCID)** element for this property.

LCID: Indicates localization information. This **MUST** be a valid LCID if **UseLCID** is set to true. This parameter **MUST** be 0 if **UseLCID** is set to **false**.

2.2.4.3 ArrayOf_PortalItem

Holds an array of **_PortalItem** elements (section [2.2.4.4](#)).

```
<s:complexType name="ArrayOf_PortalItem">
  <s:sequence>
    <s:element name="_PortalItem" type="s0:_PortalItem" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

_PortalItem: An individual **_PortalItem** (section 2.2.4.4) entry.

2.2.4.4 _PortalItem

Returns the last modification time for items.

```
<s:complexType name="_PortalItem">
  <s:sequence>
    <s:element name="ID" type="s:string" minOccurs="0"/>
    <s:element name="LastModified" type="s:dateTime"/>
  </s:sequence>
```

</s:complexType>

ID: The identifier of the item that was requested. This element **MUST** be present.

LastModified: The last modification time of the item identified by the **ID** element.

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

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is 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 Server Details

3.1.1 Abstract Data Model

This protocol supports reading items that conform to a hierarchical pattern of data organization. This section specifies the hierarchical pattern in detail.

The elements of this hierarchy include a service application or a site(2) and items. The data organization includes two more hierarchy levels in the data model, one for the search folders (1), and one for the buckets (1). These elements conform to the data organization shown in the following diagram.

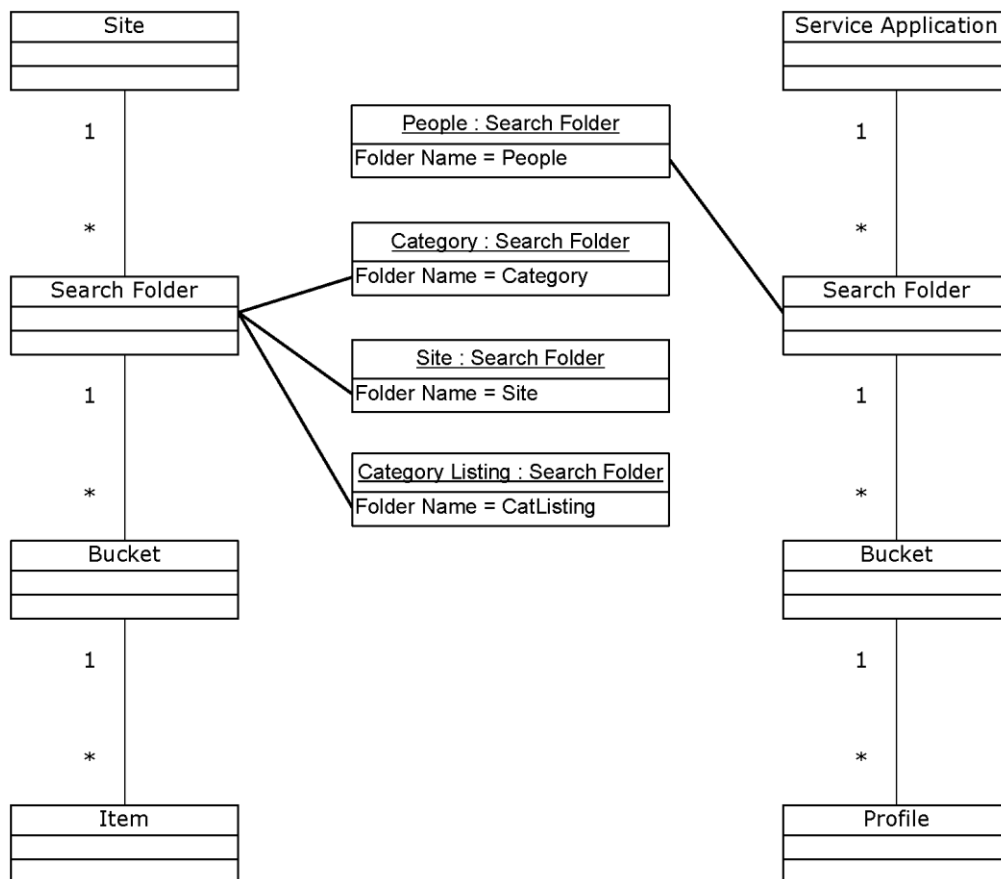


Figure 4: Abstract data model organization

3.1.2 Timers

None.

3.1.3 Initialization

The following initialization steps MUST be performed:

1. All abstract data model entities contain data that MUST be retrieved from persistent storage at initialization time.
2. Each operation within the protocol server MUST begin listening for requests at the respective URL addresses given in the message transport (as specified in section [2.1](#)).

3.1.4 Message Processing Events and Sequencing Rules

Operation	Description
EnumerateBucket	Used to request a list of items in a bucket(1).
1. EnumerateFolder	Used to request a list of buckets(1), in each partition , available in a search folder (1).
2. GetBucket	Used to request data for the bucket(1).
3. GetItem	Used to request property values of the item.
4. GetSite	Used to request properties of the current site (2).

3.1.4.1 EnumerateBucket

This operation is used to request a list of items from the bucket (1).

```
<operation name="EnumerateBucket">
  <input message="s0:EnumerateBucketSoapIn"/>
  <output message="s0:EnumerateBucketSoapOut"/>
</operation>
```

The protocol client sends an **EnumerateBucketSoapIn** (section [3.1.4.1.1.1](#)) request message and the protocol server responds with an **EnumerateBucketSoapOut** (section [3.1.4.1.1.2](#)) response message.

3.1.4.1.1 Messages

3.1.4.1.1.1 EnumerateBucketSoapIn

This message is the request message for **EnumerateBucket** (Section [3.1.4.1](#)).

The **SOAP action** value of the message is defined as follows:

```
http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/EnumerateBucket
```


The **SOAP body** contains an **EnumerateBucket** element (section [3.1.4.1.2.1](#)).

3.1.4.1.1.2 EnumerateBucketSoapOut

This message is the response message for **EnumerateBucket** (Section [3.1.4.1](#)).

The SOAP body contains an **EnumerateBucketResponse** (section [3.1.4.1.2.2](#)) element.

3.1.4.1.2 Elements

3.1.4.1.2.1 EnumerateBucket

The **EnumerateBucket** element is used to issue the **EnumerateBucket** request.

```
<s:element name="EnumerateBucket">
  <s:complexType>
    <s:sequence>
      <s:element name="strFolder" type="s:string" minOccurs="0"/>
      <s:element name="strBucketId" type="s:string" minOccurs="0"/>
      <s:element name="strCatalog" type="s:string" minOccurs="0"/>
      <s:element name="strScope" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

strFolder: A parameter representing the name of the search folder (1) operated upon by the **crawl** process. The implementation MUST be case-insensitive. The value MUST be the string value "people".

strBucketId: A parameter representing the identifier of the bucket(1) that is crawled. The value MUST contain a 32-bit integer, culture-independent encoded as a string.

strCatalog: MUST be ignored.

strScope: MUST be ignored.

3.1.4.1.2.2 EnumerateBucketResponse

EnumerateBucketResponse is returned as the response of the **EnumerateBucket** (section [3.1.4.1](#)) request.

```
<s:element name="EnumerateBucketResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="EnumerateBucketResult" type="s:unsignedInt"/>
      <s:element name="vItems" type="s0:ArrayOf_PortalItem" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

EnumerateBucketResult: The return value of the call, which MUST be 0.

vItems: The list of items identified by the **strFolder** and **strBucketId** parameters.

3.1.4.2 EnumerateFolder

This operation is used to request a list of buckets(1) in a search folder (1).

```
<operation name="EnumerateFolder">
  <input message="s0:EnumerateFolderSoapIn"/>
```

```
<output message="s0:EnumerateFolderSoapOut"/>
</operation>
```

The protocol client sends an **EnumerateFolderSoapIn** (section [3.1.4.2.1.1](#)) request message, and the server responds with an **EnumerateFolderSoapOut** (section [3.1.4.2.1.2](#)) response message.

3.1.4.2.1 Messages

3.1.4.2.1.1 EnumerateFolderSoapIn

This message is the request message for **EnumerateFolder** (section [3.1.4.2](#)).

The SOAP action value of the message is specified as follows:

```
http://microsoft.com/webservices/SharePPointPortalServer/PortalCrawl/EnumerateFolder
```

The SOAP body contains an **EnumerateFolder** element (section [3.1.4.2.2.1](#)).

3.1.4.2.1.2 EnumerateFolderSoapOut

This message is the response message for **EnumerateFolder** (section [3.1.4.2](#)).

The SOAP body contains an **EnumerateFolderResponse** element (section [3.1.4.2.2.2](#)).

3.1.4.2.2 Elements

3.1.4.2.2.1 EnumerateFolder

The **EnumerateFolder** element is used to issue the **EnumerateFolder** (section [3.1.4.2](#)) request.

```
<s:element name="EnumerateFolder">
  <s:complexType>
    <s:sequence>
      <s:element name="strFolder" type="s:string" minOccurs="0"/>
      <s:element name="strCatalog" type="s:string" minOccurs="0"/>
      <s:element name="strScope" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

strFolder: The parameter representing the name of the search folder (1) that is crawled. Implementation MUST be case-insensitive. The value MUST be the string value "people". **strCatalog:** MUST be ignored.

strScope: MUST be ignored.

3.1.4.2.2.2 EnumerateFolderResponse

EnumerateFolderResponse is returned as the response of the **EnumerateFolder** request (Section [3.1.4.2](#)).

```
<s:element name="EnumerateFolderResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="EnumerateFolderResult" type="s:unsignedInt"/>
      <s:element name="vItems" type="s0:ArrayOf_PortalItem" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

```
</s:sequence>
</s:complexType>
</s:element>
```

EnumerateFolderResult: The return value of the call, which MUST be 0.

vItems: The list of buckets(1) available in the search folder (1) identified by the **strFolder** parameter.

3.1.4.3 GetBucket

This operation is used to request property values of the buckets (1).

The SOAP action value of the message is specified as follows:

```
<operation name="GetBucket">
  <input message="s0:GetBucketSoapIn"/>
  <output message="s0:GetBucketSoapOut"/>
</operation>
```

The protocol client sends a **GetBucketSoapIn** request message (section [3.1.4.3.1.1](#)), and the protocol server responds with a **GetBucketSoapOut** response message (section [3.1.4.3.1.2](#)).

3.1.4.3.1 Messages

3.1.4.3.1.1 GetBucketSoapIn

This message is the request message for **GetBucket** (section [3.1.4.3](#)).

The SOAP action value of the message is specified as follows:

```
http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/GetBucket
```

The SOAP body contains a **GetBucket** (section [3.1.4.3.2.1](#)) element.

3.1.4.3.1.2 GetBucketSoapOut

This message is the response message for **GetBucket** (section [3.1.4.3](#)).

The SOAP body contains a **GetBucketResponse** element (section [3.1.4.3.2.2](#)).

3.1.4.3.2 Elements

3.1.4.3.2.1 GetBucket

The **GetBucket** element is used to issue the **GetBucket** request.

```
<s:element name="GetBucket">
  <s:complexType>
    <s:sequence>
      <s:element name="strFolder" type="s:string" minOccurs="0"/>
      <s:element name="strBucketId" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

strFolder: The parameter representing the name of the search folder (1) that is crawled. The implementation MUST be case-insensitive. The value MUST be the string value "people".

strBucketId: The identifier that represents the bucket (1) that is crawled. The value MUST contain a 32-bit integer, culture-independent, and encoded as a string.

3.1.4.3.2.2 GetBucketResponse

The **GetBucketResponse** element is returned as the response of the **GetBucket** request (section [3.1.4.3](#)).

```
<s:element name="GetBucketResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetBucketResult" type="s:unsignedInt"/>
      <s:element name="vPropValues" type="s0:ArrayOf_PortalPropValue" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetBucketResult: The return value of the call, which MUST be 0.

vPropValues: An array of properties for the bucket (1) identified by the **strFolder** and **strBucketId** parameters. MUST return the values for properties described in the following table.

Property name	Property type	Description
DAV:getlastmodified	Datetime	Date of last modification of the requested bucket

3.1.4.4 GetItem

This operation is used to request properties of the items.

```
<operation name="GetItem">
  <input message="s0:GetItemSoapIn"/>
  <output message="s0:GetItemSoapOut"/>
</operation>
```

The protocol client sends a **GetItemSoapIn** (section [3.1.4.4.1.1](#)) request message, and the protocol server responds with a **GetItemSoapOut** (section [3.1.4.4.1.2](#)) response message.

3.1.4.4.1 Messages

3.1.4.4.1.1 GetItemSoapIn

This message is the request message for **GetItem** (section [3.1.4.4](#)).

The SOAP action value of the message is specified as follows:

```
http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/GetItem
```

The SOAP body contains a **GetItem** (section [3.1.4.4.2.1](#)) element.

3.1.4.4.1.2 GetItemSoapOut

This message is the response message for the **GetItem** (section [3.1.4.4](#)).

The SOAP body contains a **GetItemResponse** (section [3.1.4.4.2.2](#)) element.

3.1.4.4.2 Elements

3.1.4.4.2.1 GetItem

The **GetItem** element is used to issue the **GetItem** request (section [3.1.4.4](#)).

```
<s:element name="GetItem">
  <s:complexType>
    <s:sequence>
      <s:element name="strFolder" type="s:string" minOccurs="0"/>
      <s:element name="strItemId" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

strFolder: A parameter representing the name of the search folder (1) that is crawled. The implementation MUST be case-insensitive. This value MUST be the string value "people".

strItemId: This value MUST be the identifier of the item in the search folder (1) provided in the **strFolder** parameter.

3.1.4.4.2.2 GetItemResponse

The **GetItemResponse** element is returned as the response of the **GetItem** request (section [3.1.4.4](#)).

```
<s:element name="GetItemResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetItemResult" type="s:unsignedInt"/>
      <s:element name="vPropValues" type="s0:ArrayOf_PortalPropValue" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetItemResult: The return value of the call, which MUST be 0.

vPropValues: The array of properties for the item identified by **strFolder** and **strItemId** parameters. MUST return values for the properties described in the following table.

Property name	Property type	Description
DAV:getlastmodified	datetime	The date of last modification of the item requested.
DAV:href	string	Set to the item URL.
DAV:contentclass	string	SHOULD<1> be set to "urn:content-class:SPSPeople". It MAY<2> instead be one of the following values: <ul style="list-style-type: none">urn:content-classes:SPSSiteRegistryurn:content-classes:SPSSiteListing

Property name	Property type	Description
		<ul style="list-style-type: none"> urn:content-classes:SPSListing

MAY<3> return the property described in the following table.

Property name	Property type	Description
urn:schemas-microsoft-com:sharepoint:portal:objctid	string	Set to the object identifier of the requested item.

3.1.4.5 GetSite

This operation is used to request properties of the current site (2).

```
<operation name="GetSite">
  <input message="s0:GetSiteSoapIn"/>
  <output message="s0:GetSiteSoapOut"/>
</operation>
```

The protocol client sends a **GetSiteSoapIn** request message (section [3.1.4.5.1.1](#)), and the protocol server responds with a **GetSiteSoapOut** response message (section [3.1.4.5.1.2](#)).

3.1.4.5.1 Messages

3.1.4.5.1.1 GetSiteSoapIn

This message is the request message for **GetSite** (section [3.1.4.5](#)).

The SOAP action value of the message is specified as follows:

```
http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/GetSite
```

The SOAP body contains a **GetSite** request (section [3.1.4.5.2.1](#)) element.

3.1.4.5.1.2 GetSiteSoapOut

This message is the response message for **GetSite** (section [3.1.4.5](#)).

The SOAP body contains a **GetSiteResponse** (section [3.1.4.5.2.2](#)) element.

3.1.4.5.2 Elements

3.1.4.5.2.1 GetSite

The **GetSite** element is used to issue the **GetSite** request.

```
<s:element name="GetSite">
```

```

<s:complexType>
  <s:sequence>
    <s:element name="sSite" type="s0:_PortalSite"/>
  </s:sequence>
</s:complexType>
</s:element>

```

sSite: Contains input site data.

3.1.4.5.2.2 GetSiteResponse

The **GetSiteResponse** element is returned as the response of the **GetSite** (section [3.1.4.5](#)) request.

```

<s:element name="GetSiteResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetSiteResult" type="s:unsignedInt"/>
      <s:element name="sSite" type="s0:_PortalSite"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

GetSiteResult: The return value of the call, which MUST be 0.

sSite: Output site information. The call MUST replace the existing values of **strSiteId**, **iLocale**, **strSiteName**, **strApplicationId** and **strPartitionId**.

3.1.4.5.3 Complex Types

3.1.4.5.3.1 _PortalSite

Used to request and retrieve information about a site (2).

```

<s:complexType name="_PortalSite">
  <s:sequence>
    <s:element name="strSiteId" type="s:string" minOccurs="0"/>
    <s:element name="strPortalUrl" type="s:string" minOccurs="0"/>
    <s:element name="strPortalUrlLastModified" type="s:string" minOccurs="0"/>
    <s:element name="iLocale" type="s:int"/>
    <s:element name="strSiteName" type="s:string" minOccurs="0"/>
    <s:element name="strApplicationId" type="s:string" minOccurs="0"/>
    <s:element name="strPartitionId" type="s:string" minOccurs="0"/>
  </s:sequence>
</s:complexType>

```

strSiteId: The identifier of the site (2).

strPortalUrl: The URL of the site (2). This value MUST be ignored.

strPortalUrlLastModified: The date and time of last modification of the content of the site (2). This value MUST be ignored.

iLocale: Contains the language code identifier (LCID) of the site (2).

strSiteName: Contains name of the site (2).

strApplicationId: The identifier of the application.

strPartitionId: The identifier of the partition (1).

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 People Search

This protocol supports reading user profiles.

4.1.1 Data

The following data supports the crawling of user profiles for a site (2).

The bucket (1) size for user profiles is 10,000.

The **user profile store** contains two buckets (1), Bucket1 and Bucket2, which store 20,000 user profiles representing different directory accounts:

- Bucket1 – contains user profiles UP1, UP2,..., UP10000
- Bucket2 – contains user profiles UP10001, UP10002,...,UP20000

4.1.2 Full Crawl

For a full crawl, the protocol client loops through the entire collection of user profiles and retrieves property values needed for the search indexer. To do this, the protocol client follows these steps:

- Call the **GetSite** (section [3.1.4.5](#)) method to retrieve information about the Web site that will be crawled, passing:
 - **sSite**: the reference to a **_PortalSite** object section [3.1.4.5.3.1](#)).

The protocol client sends a **GetSiteSoapIn** request message (section [3.1.4.5.1.1](#)), as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetSite xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <sSite>
        <iLocale>0</iLocale>
      </sSite>
    </GetSite>
  </soap:Body>
</soap:Envelope>
```

Assuming no errors occurred, the returned **_PortalSite** object (section 3.1.4.5.3.1) has information about the site (2) being crawled:

- **strSiteId** = "d8b5812e-b7e9-481b-92ea-78f029fde92f"
- **strSiteName** = "Central Administration"
- **iLocale** = 1033
- **strApplicationId** = "6b29fc40-ca47-1067-b31d-00dd010662da"
- **strPartitionId** = "936da01f-9abd-4d9d-80c7-02af85c822a8"

The protocol server responds with a **GetSiteSoapOut** response message (section [3.1.4.5.1.2](#)) response message, as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<GetSiteResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
<GetSiteResult>0</GetSiteResult>
<sSite>
<strSiteId>d8b5812e-b7e9-481b-92ea-78f029fde92f</strSiteId>
<iLocale>1033</iLocale>
<strSiteName>Central Administration</strSiteName>
<strApplicationId>6b29fc40-ca47-1067-b31d-00dd010662da</strApplicationId>
<strPartitionId>936da01f-9abd-4d9d-80c7-02af85c822a8</strPartitionId>
</sSite>
</GetSiteResponse>
</soap:Body>
</soap:Envelope>
```

- Call the **EnumerateFolder** (section [3.1.4.2.2.1](#)) method to retrieve the list of buckets(1) available in a particular search folder (1), and pass in the following parameter:
 - **strFolder:** Set to "people", as the value for the search folder (1) being crawled.

The protocol client sends an **EnumerateFolderSoapIn** request message (section [3.1.4.2.1.1](#)), as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<EnumerateFolder
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
<strFolder>people</strFolder>
</EnumerateFolder>
</soap:Body>
</soap:Envelope>
```

Assuming no errors occurred, vItems is the returned **ArrayOf_PortalItem** object (section [2.2.4.3](#)), which contains the available buckets (1) as follows:

- vItems[0] contains information about bucket (1) Bucket1:
 - ID = "1"
 - LastModified = {1/29/2008 11:20:04 PM}
- vItems[1] contains information about bucket (1) Bucket2:
 - ID = "2"
 - LastModified = {1/29/2008 11:20:08 PM}

The protocol server responds with an **EnumerateFolderSoapOut** (section [3.1.4.2.1.2](#)) response message, as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<EnumerateFolderResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
```

```

<EnumerateFolderResult>0</EnumerateFolderResult>
<vItems>
  <_PortalItem>
    <ID>1</ID>
    <LastModified>2008-01-29T23:20:04.077</LastModified>
  </_PortalItem>
  <_PortalItem>
    <ID>2</ID>
    <LastModified>2008-01-29T19:20:08.263</LastModified>
  </_PortalItem>
</vItems>
</EnumerateFolderResponse>
</soap:Body>
</soap:Envelope>

```

For the full crawl example, the protocol client crawls all buckets (1) returned by **EnumerateFolder** (section [3.1.4.2](#)), repeating the next two steps for each bucket(1).

- Call the **GetBucket** request (section [3.1.4.3](#)) to retrieve the last modified date of Bucket Bucket1, passing:
 - **strFolder**: Set to "people", as the value for the search folder (1) being crawled
 - **strBucketId**: "1", as the Bucket Bucket1 identifier

The protocol client sends an **GetBucketSoapIn** request message (section [3.1.4.3.1.1](#)), as follows.

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetBucket xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <strFolder>people</strFolder>
      <strBucketId>1</strBucketId>
    </GetBucket>
  </soap:Body>
</soap:Envelope>

```

Assuming no errors occurred; **vPropValues** is the returned **ArrayOf_PortalPropValue** (section [2.2.4.1](#)) object containing the available information for Bucket Bucket1, as follows.

- vPropValues[0]:
 - Count = 1
 - LCID = 0
 - Type = "datetime"
 - URI = "DAV:getlastmodified"
 - UseLCID = false
 - Value = " 2008-02-21 19:54:37Z"

The protocol server responds with a **GetBucketSoapOut** response message (section [3.1.4.3.1.2](#)), as follows.

```

<?xml version="1.0" encoding="utf-8" ?>

```

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetBucketResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <GetBucketResult>0</GetBucketResult>
      <vPropValues>
        <_PortalPropValue>
          <URI>DAV:getlastmodified</URI>
          <Value>2008-02-21 19:54:37Z</Value>
          <Type>datetime</Type>
          <Count>1</Count>
          <UseLCID>>false</UseLCID>
          <LCID>0</LCID>
        </_PortalPropValue>
      </vPropValues>
    </GetBucketResponse>
  </soap:Body>
</soap:Envelope>

```

- Call **EnumerateBucket** (section [3.1.4.1](#)) to retrieve the list of user profiles in bucket(1) Bucket1, passing:
 - **strFolder**: Set to "people", as the value for the search folder (1) being crawled.
 - **strBucketId**: Set to 1, as the Bucket1 identifier.

The protocol client sends an **EnumerateBucketSoapIn** request message (section [3.1.4.1.1.1](#)), as follows.

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <EnumerateBucket
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <strFolder>people</strFolder>
      <strBucketId>1</strBucketId>
    </EnumerateBucket>
  </soap:Body>
</soap:Envelope>

```

Assuming no errors occurred, **vItems** is the returned **ArrayOf_PortalItem** object (section 2.2.4.3) containing the available user profiles in Bucket1, as follows.

- vItems[0] contains information about user profile UP1:
 - ID = "1"
 - LastModified = {1/26/2008 1:05:27 AM}
- vItems[1]... vItems[9998]
- vItems[9999] contains information about user profile UP10000:
 - ID = "10000"
 - LastModified = {1/26/2008 1:17:56 AM}

The protocol server responds with an **EnumerateBucketSoapOut** (section [3.1.4.1.1.2](#)) response message, as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <EnumerateBucketResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <EnumerateBucketResult>0</EnumerateBucketResult>
      <vItems>
        <_PortalItem>
          <ID>1</ID>
          <LastModified>2008-01-26T01:05:27.97</LastModified>
        </_PortalItem>
        .....
        < PortalItem>
          <ID>10000</ID>
          <LastModified>2008-01-26T01:17:56.03</LastModified>
        </ PortalItem>
      </vItems>
    </EnumerateBucketResponse>
  </soap:Body>
</soap:Envelope>
```

For the full crawl example, the protocol client crawls all user profiles returned by **EnumerateBucket** (section 3.1.4.1), repeating the next step for each user profile.

- Call **GetItem** (section [3.1.4.4](#)) to retrieve the list of properties of the user profile UP1, passing:
 - **strFolder**: Set to "people", as the value for the search folder (1) being crawled.
 - **strItemId**: "1", as the user profile UP1 identifier.

The protocol client sends a **GetItemSoapIn** (section [3.1.4.4.1.1](#)) request message, as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetItem xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <strFolder>people</strFolder>
      <strItemId>1</strItemId>
    </GetItem>
  </soap:Body>
</soap:Envelope>
```

Assuming no errors occurred, **vPropValues** is the returned **ArrayOf_PortalPropValue** object (section 2.2.4.1) containing the available properties of the user profile UP1, such as:

- vPropValues[0] contains the following properties:
 - Count = 1
 - LCID = 0
 - Type = "datetime"
 - URI = "DAV:getlastmodified"

- UseLCID = false
- Value = "2008-01-26 01:05:27Z"
- vPropValues[1] contains the following properties:
 - Count = 1
 - LCID = 0
 - Type = "string"
 - URI = "DAV:contentclass"
 - UseLCID = false
 - Value = "urn:content-class:SPSPeople"
- vPropValues[2] contains the following properties and an associated **GUID** value.
 - Count = 1
 - LCID = 0
 - Type = "string"
 - URI = "urn:schemas-microsoft-com:sharepoint:portal:objectid"
 - UseLCID = false
 - Value = "2FA3D133-031B-427A-AFCE-B452F757DD7A"
- vPropValues[3] contains the following properties and GUID value.
 - Count = 1
 - LCID = 0
 - Type = "string"
 - URI = "DAV:href"
 - UseLCID = false
 - Value =
"http://howdytestvm:81/Person.aspx?guid=2FA3D133%2D031B%2D427A%2DAFCE%2DB452F757DD7A"

The protocol server responds with a **GetItemSoapOut** response message (section [3.1.4.4.1.2](#)), as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetItemResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <GetItemResult>0</GetItemResult>
      <vPropValues>
        <_PortalPropValue>
          <URI>DAV:getlastmodified</URI>
          <Value>2008-01-26 01:05:27Z</Value>
          <Type>datetime</Type>
```

```
<Count>1</Count>
<UseLCID>>false</UseLCID>
<LCID>0</LCID>
</_PortalPropValue>
<_PortalPropValue>
<URI>DAV:contentclass</URI>
<Value>urn:content-class:SPSPeople</Value>
<Type>string</Type>
<Count>1</Count>
<UseLCID>>false</UseLCID>
<LCID>0</LCID>
</_PortalPropValue>
<_PortalPropValue>
<URI>urn:schemas-microsoft-com:sharepoint:portal:objectid</URI>
<Value>2FA3D133-031B-427A-AFCE-B452F757DD7A</Value>
<Type>string</Type>
<Count>1</Count>
<UseLCID>>false</UseLCID>
<LCID>0</LCID>
</_PortalPropValue>
<_PortalPropValue>
<URI>DAV:href</URI>
<Value>http://howdytestvm:81/Person.aspx?guid=2FA3D133%2D031B%2D427A%2DAFCE%2DB452F757DD7A</Value>
<Type>string</Type>
<Count>1</Count>
<UseLCID>>false</UseLCID>
<LCID>0</LCID>
</_PortalPropValue>
</vPropValues>
</GetItemResponse>
</soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

This protocol supports the **Security Support Provider Interface (SSPI)** and **SQL authentication** with the protocol server role. These authentication methods are described in [\[MS-TDS\]](#).

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/">
      <s:element name="GetItem">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="strFolder" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="strItemId" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetItemResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetItemResult" type="s:unsignedInt" />
            <s:element minOccurs="0" maxOccurs="1" name="vPropValues"
type="s0:ArrayOf_PortalPropValue" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOf_PortalPropValue">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="_PortalPropValue"
type="s0:_PortalPropValue" />
        </s:sequence>
      </s:complexType>
      <s:complexType name=" PortalPropValue">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="URI" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Value" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Type" type="s:string" />
          <s:element name="Count" type="s:int" />
          <s:element name="UseLCID" type="s:boolean" />
          <s:element name="LCID" type="s:unsignedInt" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetBucket">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="strFolder" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="strBucketId" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetBucketResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetBucketResult" type="s:unsignedInt" />
            <s:element minOccurs="0" maxOccurs="1" name="vPropValues"
type="s0:ArrayOf_PortalPropValue" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="EnumerateBucket">
```

```

    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="strFolder" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="strBucketId" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="strCatalog" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="strScope" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="EnumerateBucketResponse">
    <s:complexType>
      <s:sequence>
        <s:element name="EnumerateBucketResult" type="s:unsignedInt" />
        <s:element minOccurs="0" maxOccurs="1" name="vItems" type="s0:ArrayOf_PortalItem"
/>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOf_PortalItem">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="_PortalItem"
type="s0:_PortalItem" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="_PortalItem">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="ID" type="s:string" />
        <s:element name="LastModified" type="s:dateTime" />
      </s:sequence>
    </s:complexType>
    <s:element name="EnumerateFolder">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="strFolder" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strCatalog" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strScope" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="EnumerateFolderResponse">
      <s:complexType>
        <s:sequence>
          <s:element name="EnumerateFolderResult" type="s:unsignedInt" />
          <s:element minOccurs="0" maxOccurs="1" name="vItems" type="s0:ArrayOf_PortalItem"
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSite">
        <s:complexType>
          <s:sequence>
            <s:element name="sSite" type="s0:PortalSite" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="_PortalSite">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="strSiteId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strPortalUrl" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strPortalUrlLastModified"
type="s:string" />
          <s:element name="iLocale" type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="strSiteName" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="strApplicationId" type="s:string"/>
          <s:element minOccurs="0" maxOccurs="1" name="strPartitionId" type="s:string"/>
        </s:sequence>
      </s:complexType>
    </s:element name="GetSiteResponse">

```

```

        <s:complexType>
          <s:sequence>
            <s:element name="GetSiteResult" type="s:unsignedInt" />
            <s:element name="sSite" type="s0:_PortalSite" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </types>
  <message name="GetItemSoapIn">
    <part name="parameters" element="s0:GetItem" />
  </message>
  <message name="GetItemSoapOut">
    <part name="parameters" element="s0:GetItemResponse" />
  </message>
  <message name="GetBucketSoapIn">
    <part name="parameters" element="s0:GetBucket" />
  </message>
  <message name="GetBucketSoapOut">
    <part name="parameters" element="s0:GetBucketResponse" />
  </message>
  <message name="EnumerateBucketSoapIn">
    <part name="parameters" element="s0:EnumerateBucket" />
  </message>
  <message name="EnumerateBucketSoapOut">
    <part name="parameters" element="s0:EnumerateBucketResponse" />
  </message>
  <message name="EnumerateFolderSoapIn">
    <part name="parameters" element="s0:EnumerateFolder" />
  </message>
  <message name="EnumerateFolderSoapOut">
    <part name="parameters" element="s0:EnumerateFolderResponse" />
  </message>
  <message name="GetSiteSoapIn">
    <part name="parameters" element="s0:GetSite" />
  </message>
  <message name="GetSiteSoapOut">
    <part name="parameters" element="s0:GetSiteResponse" />
  </message>
  <portType name="PortalCrawlSoap">
    <operation name="GetItem">
      <input message="s0:GetItemSoapIn" />
      <output message="s0:GetItemSoapOut" />
    </operation>
    <operation name="GetBucket">
      <input message="s0:GetBucketSoapIn" />
      <output message="s0:GetBucketSoapOut" />
    </operation>
    <operation name="EnumerateBucket">
      <input message="s0:EnumerateBucketSoapIn" />
      <output message="s0:EnumerateBucketSoapOut" />
    </operation>
    <operation name="EnumerateFolder">
      <input message="s0:EnumerateFolderSoapIn" />
      <output message="s0:EnumerateFolderSoapOut" />
    </operation>
    <operation name="GetSite">
      <input message="s0:GetSiteSoapIn" />
      <output message="s0:GetSiteSoapOut" />
    </operation>
  </portType>
  <binding name="PortalCrawlSoap" type="s0:PortalCrawlSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="GetItem">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/GetItem"
style="document" />
      <input>
        <soap:body use="literal" />

```

```

        </input>
      </output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="GetBucket">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/GetBucket"
      style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="EnumerateBucket">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/EnumerateBucket"
      style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="EnumerateFolder">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/EnumerateFolder"
      style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="GetSite">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/PortalCrawl/GetSite"
      style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
</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
- Microsoft Search Server 2010
- Microsoft Office SharePoint Portal Server 2003
- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016

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

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

[<1> Section 3.1.4.4.2.2](#): When Microsoft SharePoint Server 2010 crawls servers running SharePoint Portal Server 2003, the valid values for this property are described in the following table.

Value	Description
urn:content-classes:SPSSiteRegistry	The Item is a category .
urn:content-classes:SPSSiteListing	The Item is a site (2).
urn:content-classes:SPSListing	The Item is a category listing.

[<2> Section 3.1.4.4.2.2](#): When SharePoint Server 2010 crawls servers running SharePoint Portal Server 2003, the valid values for this property are described in the following table.

Value	Description
urn:content-classes:SPSSiteRegistry	The Item is a category.
urn:content-classes:SPSSiteListing	The Item is a site (2).
urn:content-classes:SPSListing	The Item is a category listing.

[<3> Section 3.1.4.4.2.2](#): Office SharePoint Server 2007 returns the **urn:schemas-microsoft-com:sharepoint:portal:objectid** property, SharePoint Server 2010 does not.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Z Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

9 Index

—

[PortalItem complex type](#) 13
[PortalPropValue complex type](#) 12

A

Abstract data model
[server](#) 15
[Applicability](#) 10
[ArrayOf PortalItem complex type](#) 13
[ArrayOf PortalPropValue complex type](#) 12
[Attribute groups](#) 14
[Attributes](#) 14

C

[Capability negotiation](#) 10
[Change tracking](#) 37
[Complex types](#) 12
[PortalItem](#) 13
[PortalPropValue](#) 12
[ArrayOf PortalItem](#) 13
[ArrayOf PortalPropValue](#) 12

D

Data model - abstract
[server](#) 15

E

Events
[local - server](#) 23
[timer - server](#) 23
Examples
[people search](#) 24

F

[Fields - vendor-extensible](#) 10
[Full WSDL](#) 32

G

[Glossary](#) 6
[Groups](#) 14

I

[Implementer - security considerations](#) 31
[Index of security parameters](#) 31
[Informative references](#) 8
Initialization
[server](#) 16
[Introduction](#) 6

L

Local events

[server](#) 23

M

Message processing
[server](#) 16
Messages
[PortalItem complex type](#) 13
[PortalPropValue complex type](#) 12
[ArrayOf PortalItem complex type](#) 13
[ArrayOf PortalPropValue complex type](#) 12
[attribute groups](#) 14
[attributes](#) 14
[complex types](#) 12
[elements](#) 12
[enumerated](#) 11
[groups](#) 14
[namespaces](#) 11
[simple types](#) 14
[syntax](#) 11
[transport](#) 11

N

[Namespaces](#) 11
[Normative references](#) 7

O

Operations
[EnumerateBucket](#) 16
[EnumerateFolder](#) 17
[GetBucket](#) 19
[GetItem](#) 20
[GetSite](#) 22
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 31
[People search example](#) 24
[example data](#) 24
[full crawl](#) 24
[Preconditions](#) 10
[Prerequisites](#) 10
[Product behavior](#) 36
Protocol Details
[overview](#) 15

R

[References](#) 7
[informative](#) 8
[normative](#) 7
[Relationship to other protocols](#) 9

S

Security
[implementer considerations](#) 31
[parameter index](#) 31

Sequencing rules

[server](#) 16

Server

[abstract data model](#) 15

[EnumerateBucket operation](#) 16

[EnumerateFolder operation](#) 17

[GetBucket operation](#) 19

[GetItem operation](#) 20

[GetSite operation](#) 22

[initialization](#) 16

[local events](#) 23

[message processing](#) 16

[sequencing rules](#) 16

[timer events](#) 23

[timers](#) 16

[Simple types](#) 14

[Standards assignments](#) 10

Syntax

[messages - overview](#) 11

T

Timer events

[server](#) 23

Timers

[server](#) 16

[Tracking changes](#) 37

[Transport](#) 11

Types

[complex](#) 12

[simple](#) 14

V

[Vendor-extensible fields](#) 10

[Versioning](#) 10

W

[WSDL](#) 32