

[MS-ASWS]:

Access Services 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
7/13/2009	0.1	Major	Initial Availability
8/28/2009	0.2	Editorial	Revised and edited the technical content
11/6/2009	0.3	Editorial	Revised and edited the technical content
2/19/2010	1.0	Major	Updated and revised the technical content
3/31/2010	1.01	Editorial	Revised and edited the technical content
4/30/2010	1.02	Editorial	Revised and edited the technical content
6/7/2010	1.03	Minor	Updated the technical content
6/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.0	Major	Significantly changed the technical content.
6/10/2011	2.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.1	Minor	Clarified the meaning of the technical content.
4/11/2012	2.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.1	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.2	Minor	Clarified the meaning of the technical content.
11/18/2013	2.2	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.2	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.2	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/31/2014	2.2	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.2	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	3.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	8
1.1	Glossary	8
1.2	References	10
1.2.1	Normative References	10
1.2.2	Informative References	11
1.3	Overview	11
1.4	Relationship to Other Protocols	12
1.5	Prerequisites/Preconditions	12
1.6	Applicability Statement	12
1.7	Versioning and Capability Negotiation	13
1.8	Vendor-Extensible Fields	13
1.9	Standards Assignments	13
2	Messages	14
2.1	Transport	14
2.2	Common Message Syntax	14
2.2.1	Namespaces	14
2.2.2	Messages	15
2.2.2.1	Faults	15
2.2.3	Elements	15
2.2.4	Complex Types	15
2.2.4.1	ArrayOfKeyValuePair	15
2.2.4.2	KeyValuePair	15
2.2.4.3	VersionType	16
2.2.5	Simple Types	16
2.2.5.1	UpdateCommand	16
2.2.6	Attributes	17
2.2.7	Groups	17
2.2.8	Attribute Groups	17
3	Protocol Details	18
3.1	Server Details	18
3.1.1	Abstract Data Model	18
3.1.1.1	Access Services Site Template	18
3.1.1.1.1	MSysASO	19
3.1.1.1.1.1	ID	19
3.1.1.1.1.2	Title	19
3.1.1.1.1.3	owshiddenversion	19
3.1.1.1.1.4	Type	19
3.1.1.1.1.5	Revision	21
3.1.1.1.1.6	ClientObject	21
3.1.1.1.1.7	ServerObject	23
3.1.1.1.1.8	ClientObjectProperties	24
3.1.1.1.1.9	Flags	24
3.1.1.1.1.10	Attachments	24
3.1.1.1.2	USysApplicationLog	25
3.1.1.1.2.1	ID	25
3.1.1.1.2.2	Created	25
3.1.1.1.2.3	owshiddenversion	26
3.1.1.1.2.4	Category	26
3.1.1.2	Access Services Site Version	26
3.1.2	Timers	27
3.1.3	Initialization	27
3.1.4	Message Processing Events and Sequencing Rules	27
3.1.4.1	GetAccessServicesVersion	27

3.1.4.1.1	Messages	28
3.1.4.1.1.1	GetAccessServicesVersionSoapIn	28
3.1.4.1.1.2	GetAccessServicesVersionSoapOut	28
3.1.4.1.2	Elements	28
3.1.4.1.2.1	GetAccessServicesVersion	29
3.1.4.1.2.2	GetAccessServicesVersionResponse	29
3.1.4.1.3	Complex Types	29
3.1.4.1.4	Simple Types	29
3.1.4.1.5	Attributes	29
3.1.4.1.6	Groups	29
3.1.4.1.7	Attribute Groups	29
3.1.4.2	GetCurrentUserInfo	30
3.1.4.2.1	Messages	30
3.1.4.2.1.1	GetCurrentUserInfoSoapIn	30
3.1.4.2.1.2	GetCurrentUserInfoSoapOut	30
3.1.4.2.2	Elements	30
3.1.4.2.2.1	GetCurrentUserInfo	31
3.1.4.2.2.2	GetCurrentUserInfoResponse	31
3.1.4.2.3	Complex Types	31
3.1.4.2.4	Simple Types	31
3.1.4.2.5	Attributes	31
3.1.4.2.6	Groups	31
3.1.4.2.7	Attribute Groups	31
3.1.4.3	GetDataMacroState	32
3.1.4.3.1	Messages	32
3.1.4.3.1.1	GetDataMacroStateSoapIn	32
3.1.4.3.1.2	GetDataMacroStateSoapOut	32
3.1.4.3.2	Elements	32
3.1.4.3.2.1	GetDataMacroState	33
3.1.4.3.2.2	GetDataMacroStateResponse	33
3.1.4.3.3	Complex Types	33
3.1.4.3.3.1	DataMacroInstanceState	34
3.1.4.3.4	Simple Types	34
3.1.4.3.4.1	DataMacroState	34
3.1.4.3.5	Attributes	35
3.1.4.3.6	Groups	35
3.1.4.3.7	Attribute Groups	35
3.1.4.4	GetServerInformation	35
3.1.4.4.1	Messages	35
3.1.4.4.1.1	GetServerInformationSoapIn	36
3.1.4.4.1.2	GetServerInformationSoapOut	36
3.1.4.4.2	Elements	36
3.1.4.4.2.1	GetServerInformation	36
3.1.4.4.2.2	GetServerInformationResponse	36
3.1.4.4.3	Complex Types	37
3.1.4.4.3.1	AccessServerInformationType	37
3.1.4.4.4	Simple Types	37
3.1.4.4.5	Attributes	37
3.1.4.4.6	Groups	38
3.1.4.4.7	Attribute Groups	38
3.1.4.5	RunDataMacro	38
3.1.4.5.1	Messages	38
3.1.4.5.1.1	RunDataMacroSoapIn	38
3.1.4.5.1.2	RunDataMacroSoapOut	39
3.1.4.5.2	Elements	39
3.1.4.5.2.1	RunDataMacro	39
3.1.4.5.2.2	RunDataMacroResponse	39
3.1.4.5.3	Complex Types	40

3.1.4.5.4	Simple Types	40
3.1.4.5.5	Attributes	40
3.1.4.5.6	Groups	40
3.1.4.5.7	Attribute Groups.....	40
3.1.4.6	SetAccessServicesVersion	40
3.1.4.6.1	Messages	40
3.1.4.6.1.1	SetAccessServicesVersionSoapIn	41
3.1.4.6.1.2	SetAccessServicesVersionSoapOut	41
3.1.4.6.2	Elements	41
3.1.4.6.2.1	SetAccessServicesVersion	41
3.1.4.6.2.2	SetAccessServicesVersionResponse	41
3.1.4.6.3	Complex Types	42
3.1.4.6.4	Simple Types	42
3.1.4.6.5	Attributes	42
3.1.4.6.6	Groups.....	42
3.1.4.6.7	Attribute Groups.....	42
3.1.4.7	StartCompilation.....	42
3.1.4.7.1	Messages	42
3.1.4.7.1.1	StartCompilationSoapIn.....	43
3.1.4.7.1.2	StartCompilationSoapOut.....	43
3.1.4.7.2	Elements.....	43
3.1.4.7.2.1	StartCompilation.....	43
3.1.4.7.2.2	StartCompilationResponse	43
3.1.4.7.3	Complex Types	43
3.1.4.7.4	Simple Types	44
3.1.4.7.5	Attributes	44
3.1.4.7.6	Groups.....	44
3.1.4.7.7	Attribute Groups.....	44
3.1.4.8	UpdateLists	44
3.1.4.8.1	Messages	44
3.1.4.8.1.1	UpdateListsSoapIn	45
3.1.4.8.1.2	UpdateListsSoapOut.....	45
3.1.4.8.2	Elements.....	45
3.1.4.8.2.1	UpdateLists	45
3.1.4.8.2.2	UpdateListsResponse.....	48
3.1.4.8.3	Complex Types	48
3.1.4.8.3.1	Update	49
3.1.4.8.3.2	FieldValue.....	50
3.1.4.8.3.3	UpdateListsResultInfo.....	50
3.1.4.8.4	Simple Types	51
3.1.4.8.5	Attributes	51
3.1.4.8.6	Groups.....	51
3.1.4.8.7	Attribute Groups.....	51
3.1.5	Timer Events.....	51
3.1.6	Other Local Events.....	51
4	Protocol Examples	52
4.1	GetCurrentUserInfo	52
4.2	Use UpdateLists to Insert Items into a List	52
4.3	Use UpdateLists to Insert Items into Two Lists with Lookup Relationships	53
4.4	RunDataMacro and GetDataMacroState.....	55
5	Security	58
5.1	Security Considerations for Implementers	58
5.2	Index of Security Parameters	58
6	Appendix A: Full WSDL	59
7	Appendix B: Product Behavior	66

8	Change Tracking	67
9	Index	69

1 Introduction

The Access Services Protocol enables a protocol client to run and monitor tasks on a server application.

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:

attachment: An external file that is included with an Internet message or associated with an item in a SharePoint list.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [\[RFC5234\]](#).

available site template: An XML-based collection of predefined or user-defined settings that are stored as a site definition configuration or a site template, and can be used when creating a site.

Coordinated Universal Time (UTC): A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC-0 (or GMT).

current user: The user who is authenticated during processing operations on a front-end web server or a back-end database server.

data culture: The language that is used to specify number formatting for data.

data macro: A component that implements application logic and enables recognition of built-in actions and tasks for list items.

database application: A set of objects, including tables, queries, forms, reports, macros, and code modules, that are stored in a database structure.

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

field: A container for metadata within a SharePoint list and associated list items.

file extension: The sequence of characters in a file's name between the end of the file's name and the last "." character. Vendors of applications choose such sequences for the applications to uniquely identify files that were created by those applications. This allows file management software to determine which application should be used to open a file.

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 (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

list: A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

list identifier: A GUID that is used to identify a **list** in a site collection.

list item: An individual entry within a SharePoint list. Each list item has a schema that maps to fields in the list that contains the item, depending on the content type of the item.

list item identifier: See item identifier.

lookup field: A field of the Lookup type that enables users to select an item from another data source.

major version: An iteration of a software component, document, or list item that is ready for a larger group to see, or has changed significantly from the previous major version. For an item on a SharePoint site, the **minor version** is always "0" (zero) for a major version.

minor version: An iteration of a software component, document, or list item that is in progress or has changed only slightly from the previous version. For an item on a SharePoint site, the minor version number is never "0" (zero) and is incremented for each new version of an item, unless a **major version** is explicitly published. When minor versioning is disabled on a SharePoint site, only major version numbers are incremented, and the minor version is always "0" (zero).

session: A representation of application data in system memory. It is used to maintain state for application data that is being manipulated or monitored on a protocol server by a user.

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.

site collection: A set of websites (1) that are in the same content database, have the same owner, and share administration settings. A site collection can be identified by a **GUID** or the URL of the **top-level site** for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.

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.

subsite: A complete website that is stored in a named subdirectory of another website. The parent website can be the top-level site of a site collection or another subsite. Also referred to as subweb.

top-level site: The first site in a site collection. All other sites within a site collection are child sites of the top-level site. The URL of the top-level site is also the URL of the site collection.

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 [\[RFC3986\]](#).

user profile: A collection of properties that pertain to a specific person or entity within a portal site.

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.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

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].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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.

[MS-ACCDT] Microsoft Corporation, "[Access Template File Format Specification](#)".

[MS-AXL] Microsoft Corporation, "[Access Application Transfer Protocol Structure Specification](#)".

[MS-LISTSWS] Microsoft Corporation, "[Lists Web Service Protocol](#)".

[MS-SITESS] Microsoft Corporation, "[Sites Web Service Protocol](#)".

[MS-UGS] Microsoft Corporation, "[UserGroup Web Service Protocol](#)".

[MS-WSSF03] Microsoft Corporation, "[Windows SharePoint Services \(WSS\): File Operations Database Communications Version 3 Protocol](#)".

[MS-WSSTS] Microsoft Corporation, "[Windows SharePoint Services](#)".

[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>

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624>

[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/>

[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

None.

1.3 Overview

This protocol enables protocol clients to manage data exposed as **lists** by a protocol server. The protocol client issues requests to a protocol server. The protocol server receives, processes, and responds to the requests of protocol clients.

The protocol provides the following sets of functionality:

- Retrieve **user profile** information: The protocol client can get information about the **current user** from the protocol server through the Web service described in this protocol.
- Manage data stored within lists: The protocol client can add, modify, delete, and retrieve data exposed as lists by a protocol server through the Web service described in this protocol.
- Trigger a **data macro**: The protocol client can trigger a data macro through the Web service described in this protocol. The protocol server will respond with a token for this data macro instance to the protocol client. To know whether the data macro has finished, or is still processing, or encountered any error, the protocol client can get the status of the data macro instance from the protocol server by using the token for the data macro instance.

- Managing the **Access Services Site Version** (section [3.1.1.2](#)): The protocol client can retrieve or attempt to set the value of the Access services site version for a **site** or **subsite**.
- Compiling the **database application** defined by a site: The protocol client can notify a protocol server to start compiling the database application defined in the **MSysASO** list (section [3.1.1.1.1](#)) of the current site or subsite from the object definitions stored within that list.

The Web services are documented in detail in section [3](#).

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:

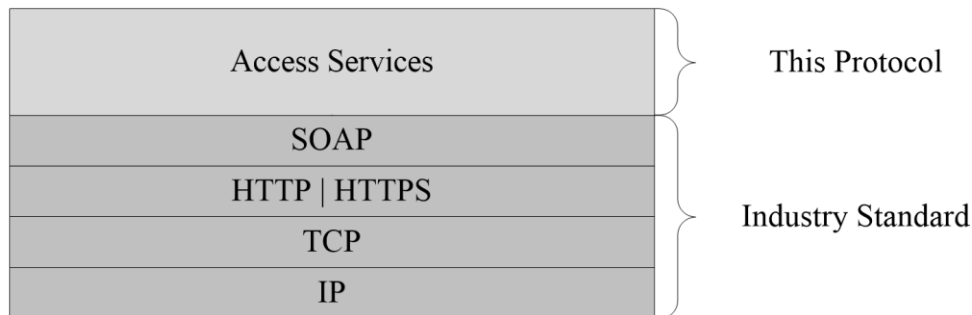


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a **URI** that is known by protocol clients. The protocol server **endpoint** is formed by appending `"/_vti_bin/ ACCSRV/AccessServer.asmx"` to the site URI. For example:

- `http://www.example.com/Repository/_vti_bin/ ACCSRV/AccessServer.asmx`

This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

The Access Services Web Services Protocol is applicable in the following scenarios:

- Retrieving user profile information about the current user.
- Running a data macro on the protocol server.
- Inserting, updating, or deleting data on the protocol server.
- Retrieving and setting the Access services site version (section [3.1.1.2](#)).
- Triggering compilation of a database application.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol can be implemented by using transports that support sending Simple Object Access Protocol (SOAP) messages, as described in section 2.1.
- **Protocol Versions:** This protocol is not versioned.

Capability Negotiation: This protocol does not support version negotiation.

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 or HTTPS.

All protocol messages **MUST** be transported by using HTTP bindings at the transport level.

Protocol messages **MUST** be formatted as specified in either [\[SOAP1.1\]](#) section 4 or [\[SOAP1.2/1\]](#) section 5. Protocol server faults **MUST** be returned by using either HTTP status codes, as specified in [\[RFC2616\]](#) section 10, or **SOAP faults**, as specified in [\[SOAP1.1\]](#) section 4.4 or [\[SOAP1.2/1\]](#) section 5.4.

If the HTTPS transport is used, a server certificate **MUST** be deployed.

This protocol **MAY** transmit an additional SOAP header, the **ServiceContext** header, as specified in [\[MS-SPSTWS\]](#).

This protocol does not define any means for activating a protocol server or protocol client. The protocol server **MUST** be configured and begin listening in an implementation-specific way. In addition, the protocol client **MUST** know the format and transport that is used by the protocol server, for example, the SOAP format over an HTTP transport.

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.

Prefix	Namespace URI	Reference
http	http://schemas.xmlsoap.org/wsd/http/	
soap	http://schemas.xmlsoap.org/wsd/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsd/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/	
wSDL	http://schemas.xmlsoap.org/wsd/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

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 [\[WSDL\]](#).

This specification does not define any common **WSDL message** definitions.

2.2.2.1 Faults

In the event of an application error, the protocol server returns a SOAP fault as a response to the operation, as specified in [\[SOAP1.1\]](#) section 4.4 or [\[SOAP1.2/1\]](#) section 5.4.

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
ArrayOfKeyValuePair	An array of KeyValuePair elements.
KeyValuePair	Specifies the name and value of a parameter, as specified in [MS-AXL] section 2.2.3.18, or a return variable, as specified in [MS-AXL] section 2.1.3.3.3, for a data macro, as specified in [MS-AXL] section 2.1.3.
VersionType	Specifies the Access Services Site Version (section 3.1.1.2) of a site or subsite formatted as XML.

2.2.4.1 ArrayOfKeyValuePair

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

An array of **KeyValuePair** elements.

```
<xs:complexType name="ArrayOfKeyValuePair" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="KeyValuePair"
type="tns:KeyValuePair"/>
  </xs:sequence>
</xs:complexType>
```

KeyValuePair: Specified in section [2.2.4.2](#).

2.2.4.2 KeyValuePair

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

Specifies the name and value of a parameter, as specified in [\[MS-AXL\]](#) section 2.2.3.18, or a return variable, as specified in [\[MS-AXL\]](#) section 2.1.3.3.3, for a data macro, as specified in [\[MS-AXL\]](#) section 2.1.3.

```
<xs:complexType name="KeyValuePair" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Key"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Value" nillable="true"/>
  </xs:sequence>
</xs:complexType>
```

Key: A value that uniquely identifies the **KeyValuePair** within a collection. **MUST** be present. **MUST** be of type **xs:string** or **xs:int**.

Value: Data associated with a given **Key**, which can have any value as long as the document remains well-formed, as specified in [\[XML\]](#) section 2. **MUST** be present. **MUST** be formatted in the **data culture** of the **session**.

2.2.4.3 VersionType

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

Specifies the **Access Services Site Version** (section [3.1.1.2](#)) of a site or subsite formatted as XML.

```
<xs:complexType name="VersionType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="Major" type="xs:int" use="required"/>
  <xs:attribute name="Minor" type="xs:int" use="required"/>
</xs:complexType>
```

Major: The **major version** of the **Access Services Site Version** (section 3.1.1.2).

Minor: The **minor version** of the **Access Services Site Version** (section 3.1.1.2).

2.2.5 Simple Types

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

Simple type	Description
UpdateCommand	The UpdateCommand simple type is used to specify whether an Update element is specifying an Update, Insert, or Delete command.

2.2.5.1 UpdateCommand

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

The **UpdateCommand** simple type is used to specify whether an **Update** element is specifying an Update, Insert, or Delete command.

```
<xs:simpleType name="UpdateCommand" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="u"/>
    <xs:enumeration value="i"/>
  </xs:restriction>
</xs:simpleType>
```



```
<xs:enumeration value="d"/>
</xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **UpdateCommand** simple type.

Value	Meaning
u	Update. The data in the Update element MUST replace the data in the list item specified by the id attribute of the Update element.
i	Insert. The data in the Update element MUST be used to insert a new item into a list.
d	Delete. The list item that has a list item identifier matching the id attribute of the Update element MUST be deleted from the list.

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 be less restrictive than 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 specifies additional restrictions 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 protocol client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the protocol 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.

Except where specified, protocol clients SHOULD interpret HTTP Status Codes returned by the protocol server as specified in [\[RFC2616\]](#) section 10.

This protocol allows protocol servers to notify protocol clients of application-level faults using SOAP faults. Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults either using HTTP Status Codes or using SOAP faults as specified previously in this section.

3.1 Server 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.

Access Services Site: A site based on the Access services site template (section [3.1.1.1](#)) that supports Access services site versions (section [3.1.1.2](#)).

Lists: Lists, as specified in [\[MS-WSSTS\]](#) section 2.1.2.7, and the list items, as specified in [\[MS-WSSTS\]](#) section 2.1.2.10, contained within them.

Data Macro: A set of built-in activities that act upon list items, as specified in the **CT_DataMacro** element specified in [\[MS-AXL\]](#) section 2.2.3.49. A protocol client can trigger a data macro indirectly through **UpdateLists** (section [3.1.4.8](#)) or explicitly by calling **RunDataMacro** (section [3.1.4.5](#)). The status of a data macro instance is obtained by calling **GetMacroState** (section [3.1.4.3](#)).

3.1.1.1 Access Services Site Template

The protocol server MUST have an **available site template** named "ACCSRV#0". This means that the protocol server MUST list "ACCSRV#0" as a template name in a response to **GetSiteTemplate**, as specified in [\[MS-SITESS\]](#) section 3.1.4.5, and accept "ACCSRV#0" in the **templateName** parameter to **CreateWeb**, as specified in [\[MS-SITESS\]](#) section 3.1.4.9.

A site created with the **ACCSRV#0** template is used for representing a database application.

All sites created with the **ACCSRV#0** template MUST contain at least two lists, one with the title "MSysASO" and one with the title "USysApplicationLog", each with schema specified as follows.

3.1.1.1.1 MSysASO

The **MSysASO** list is used to specify objects and settings of a database application. Each list item in the list specifies either a single object or a group of settings, depending on the **Type field**.

Each list item in **MSysASO** MUST have a unique combination of values in their **Title** and **Type** fields.

The **MSysASO** list MUST have at least the following fields, with the following **FieldDefinitions**, as specified in [\[MS-WSSFO3\]](#) section 2.2.7.3.3.2.

3.1.1.1.1.1 ID

Specifies a unique identifying value for each list item.

FieldDefinition property	Value
Name	ID
DisplayName	ID
Type	Counter
PrimaryKey	TRUE

3.1.1.1.1.2 Title

Specifies the name of the object in the database application that is represented by this list item.

FieldDefinition property	Value
Name	Title
DisplayName	Title
Type	Text

3.1.1.1.1.3 owshiddenversion

Specifies a value used for conflict detection, as specified in [\[MS-LISTSWS\]](#) section 3.1.4.31.2.1, and by the **UpdateLists** command specified in section [3.1.4.8](#).

FieldDefinition property	Value
Name	owshiddenversion
DisplayName	owshiddenversion
Type	Integer

3.1.1.1.1.4 Type

Specifies the type of object in the database application represented by this list item.

FieldDefinition property	Value
Name	Type
DisplayName	Type
Type	Integer

The following table specifies the valid values of the **Type** field and their meanings.

Value	Meaning
0	A list and its associated data macros, as specified in [MS-AXL] section 2.1.3.2.
1	Query, as specified in [MS-AXL] section 2.1.4.
2	Form, as specified in [MS-AXL] section 2.1.2.
3	Report, as specified in [MS-AXL] section 2.1.5.
4	User interface macro, as specified in [MS-AXL] section 2.1.3.1.
5	Client data that MUST be preserved by the protocol server.
6	Client data that MUST be preserved by the protocol server.
7	Client data that MUST be preserved by the protocol server.
8	Client data that MUST be preserved by the protocol server.
9	Client data that MUST be preserved by the protocol server.
10	Client data that MUST be preserved by the protocol server.
11	Client data that MUST be preserved by the protocol server.
12	Shared image, as specified in [MS-AXL] section 2.1.6.
13	Client data that MUST be preserved by the protocol server.
14	Client data that MUST be preserved by the protocol server.
15	This record contains server private data. Protocol clients MUST NOT change the values in this record.
16	Client data that MUST be preserved by the protocol server.
17	Client data that MUST be preserved by the protocol server.
18	Client data that MUST be preserved by the protocol server.

When the **Type** field of this list item is zero, there MUST be a list in this site where the title of the list matches the value of this list item's **Title** field. That list specifies the data for the list object represented by this list item.

When the **Type** field is one of the values in the following table, the **Title** field MUST be the value specified in the table.

Type	Title
9	Navigation Pane
10	VBA References
11	DBProps

3.1.1.1.1.5 Revision

Specifies the number of updates that have been made to this list item that have changed any of the **ClientObject**, **ServerObject**, **ClientObjectProperties**, or **Attachments** fields.

FieldDefinition property	Value
Name	Revision
DisplayName	Revision
Type	Integer

3.1.1.1.1.6 ClientObject

Specifies a client-specific object of the database application. The contents of this field are created by the client and MUST be ignored by the server.

FieldDefinition property	Value
Name	ClientObject
DisplayName	ClientObject
Type	Note
RichText	FALSE
Required	FALSE

The contents of the **ClientObject** field depend on the value in the **Type** field of this list item, as specified in the following table.

Type field value	ClientObject field contents
1	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "Query".

Type field value	ClientObject field contents
2	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "Form".
3	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "Report".
4	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "Macro".
5	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "Module".
6	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "Link".
7	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from an Object part, as specified in [MS-ACCDT] section 2.1.4.14, where the value of AccessObject/Type in the Object Metadata part, as specified in [MS-ACCDT] section 2.1.4.15, was "SQLLink".
8	The contents MUST have been specified by the client by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4).
9	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from a Navigation Pane part, as specified in [MS-ACCDT] section 2.1.4.13.
10	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4), or obtained from a Visual Basic References part, as specified in [MS-ACCDT] section 2.1.4.26.
16	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4).
17	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section

Type field value	ClientObject field contents
	3.1.4.31, or UpdateLists (section 3.1.4.4).
18	The contents MUST have either been specified by the client directly by a call to UpdateListItems , as specified in [MS-LISTSWS] section 3.1.4.31, or UpdateLists (section 3.1.4.4).

The **ClientObject** field MUST be empty for all other values of the Type field.

3.1.1.1.1.7 ServerObject

Specifies an object of the database application using [\[MS-AXL\]](#). The target XML namespaces used in all the Xml elements referenced in this section are dependent on the Access Services Site Version (section [3.1.1.2](#)) for the current site.

FieldDefinition property	Value
Name	ServerObject
DisplayName	ServerObject
Type	Note
RichText	FALSE
Required	FALSE

The contents of the **ServerObject** field depend on the value in the **Type** field of this list item, as specified in the following table.

Type field value	ServerObject field contents
0	When the list represented by this list item has data macros, as specified in [MS-AXL] section 2.1.3.2, the ServerObject field contains a DataMacros element, as specified in [MS-AXL] section 2.2.1.5, specifying those data macros belonging to this list. When the list has no data macros, the ServerObject field MUST be empty.
1	MUST be XML beginning with a Query element, as specified in [MS-AXL] section 2.2.1.2, or nothing.
2	MUST be XML beginning with a View element, as specified in [MS-AXL] section 2.2.1.7, or nothing.
3	MUST be XML beginning with a Report element, as specified in [MS-AXL] section 2.4.1.1, or nothing.
4	MUST be XML beginning with a UserInterfaceMacro element, as specified in [MS-AXL] section 2.2.1.6, or nothing.
11	MUST be XML beginning with an Application element, as specified in [MS-AXL] section 2.2.1.1, or nothing.
15	The protocol server can use this field to hold any type of string data. Protocol clients MUST NOT change the values in this record.

The **ServerObject** field MUST be empty for all other values of the **Type** field.

3.1.1.1.1.8 ClientObjectProperties

Specifies client-specific properties of an object in the database application. The contents of this field are created by the client and MUST be ignored by the server.

FieldDefinition property	Value
Name	ClientObjectProperties
DisplayName	ClientObjectProperties
Type	Note
RichText	FALSE
Required	FALSE

The **ClientObjectProperties** field MUST be empty when the value of the **Type** field in this list item is not an integer between zero and seven inclusive.

When the field is not empty, its contents MUST have either been specified by the protocol client directly by a call to **UpdateListItems** ([\[MS-LISTSWS\]](#) section 3.1.4.31) or **UpdateLists** (section [3.1.4.4](#)), or obtained from an **Object Properties** part, as specified in [\[MS-ACCDT\]](#) section 2.1.4.16.

3.1.1.1.1.9 Flags

Specifies whether the object in the database application represented by this list item is a reserved server-specific object. A value of "1" specifies that this object is a reserved server-specific object, which a protocol client MUST not make changes to. A value of zero ("0") or an empty field specifies that this is not a reserved server-specific object.

FieldDefinition property	Value
Name	Flags
DisplayName	Flags
Type	Integer
Required	FALSE

3.1.1.1.1.10 Attachments

Specifies data for images and themes used by the database application.

FieldDefinition property	Value
Name	Attachments
Type	Attachments

The contents of the **Attachments** field depend on the value in the **Type** field of this list item. If the **Type** field's value is specified in the following table, there MUST be one **attachment** in the **Attachments** field. The contents of the attachment are specified in the following table.

Type field value	Attachment
12	A file of type PNG, GIF, or JPG. (The file's name without the file extension SHOULD match the Title field of this list item, but MUST

Type field value	Attachment
	be ignored by the server.) It MUST have been uploaded by the client directly by a call to AddAttachment , as specified in [MS-LISTSWS] section 3.1.4.1, or had its contents obtained from an Image part as specified in [MS-ACCDT] section 2.1.4.6.
13	A file named "Office Theme.thmx". The contents of this file are created by the client and MUST be ignored by the server. It MUST have been uploaded by the client directly by a call to AddAttachment , as specified in [MS-LISTSWS] section 3.1.4.1, or had its contents obtained from a Theme part ,as specified in [MS-ACCDT] section 2.1.4.24.
14	A file of type PNG. (The file's name without the file extension SHOULD match the Title field of this list item, but MUST be ignored by the server.) It MUST have been uploaded by the client directly by a call to AddAttachment , as specified in [MS-LISTSWS] section 3.1.4.1, or had its contents obtained from an ImageCluster part, as specified in [MS-ACCDT] section 2.1.4.7.

The **Attachments** field MUST contain zero attachments for all other values of the **Type** field.

3.1.1.1.2 USysApplicationLog

The **USysApplicationLog** list is used to specify a log of events generated by a database application on the protocol server. Each list item in USysApplicationLog contains represents a single event.

The **USysApplicationLog** list MUST have at least the following fields, with the following **FieldDefinitions** (as specified in [\[MS-WSSFO3\]](#) section 2.2.7.3.3.2).

3.1.1.1.2.1 ID

Specifies a unique identifying value for each list item.

FieldDefinition property	Value
Name	ID
DisplayName	ID
Type	Counter
PrimaryKey	TRUE

3.1.1.1.2.2 Created

Specifies the **Coordinated Universal Time (UTC)** time in which this list item was added to the list.

FieldDefinition property	Value
Name	Created
DisplayName	Created
Type	DateTime
StorageTZ	TRUE

3.1.1.1.2.3 owshiddenversion

Specifies a value used for conflict detection, as specified in [\[MS-LISTSWS\]](#) section 3.1.4.31.2.1.

FieldDefinition property	Value
Name	owshiddenversion
DisplayName	owshiddenversion
Type	Integer

3.1.1.1.2.4 Category

Specifies the type of event represented by this list item.

FieldDefinition property	Value
Name	Category
DisplayName	Category
Type	Choice

3.1.1.2 Access Services Site Version

The **Access Services Site Version** is used to determine which target XML namespaces are used when referencing application object schemas from [\[MS-AXL\]](#). This version number MUST consist of a major version and a minor version that can be expressed either as XML as specified in section [2.2.4.3](#), or as a string as specified in the following **ABNF**:

```
access-services-site-version = major-version dot minor-version  
  
major-version = 1*3DIGIT  
  
dot = "."  
  
minor-version = 1*4DIGIT
```

If the **Access Services Site Version** is 1.2, the Xml referenced in [\[MS-AXL\]](#) MUST use the target XML namespaces listed in the following table.

MS-AXL schema	Required target namespace
AXL ([MS-AXL] section 2.2)	http://schemas.microsoft.com/office/accessservices/2009/11/application
RDL ([MS-AXL] section 2.4)	http://schemas.microsoft.com/office/accessservices/2009/11/reports
Form XAML ([MS-AXL] section 2.3.3)	http://schemas.microsoft.com/office/accessservices/2009/11/forms
WPF ([MS-AXL] section 2.3.2)	http://schemas.microsoft.com/client/2009/11

MS-AXL schema	Required target namespace
XAML 2006 ([MS-AXL] section 2.3.1)	http://schemas.microsoft.com/winfx/2009/04/xaml

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of operations as defined by this specification.

Operation	Description
GetAccessServicesVersion	The GetAccessServicesVersion command is used to get the Access Services Site Version (section 3.1.1.2) of the current site or subsite.
GetCurrentUserInfo	The GetCurrentUserInfo operation returns XML that describes the user profile being used when sending requests to the protocol server.
GetDataMacroState	The GetDataMacroState command is used to get the current state of a data macro ([MS-AXL] section 2.1.3.2).
GetServerInformation	The GetServerInformation command is used to get information about which Access Services Site Versions (section 3.1.1.2) the protocol server supports and the Access Services Site Version of the current site or subsite.
RunDataMacro	The RunDataMacro operation triggers the running of a data macro ([MS-AXL] section 2.2.1.5).
SetAccessServicesVersion	The SetAccessServicesVersion command is used to set the Access Services Site Version (section 3.1.1.2) of the current site or subsite.
StartCompilation	The StartCompilation command is used to notify a protocol server to start compiling the database application defined in the MSysASO list (section 3.1.1.1.1) of the current site or subsite from the object definitions stored within that list.
UpdateLists	The UpdateLists operation is used to insert, update, and delete list items in one or more lists.

3.1.4.1 GetAccessServicesVersion

The **GetAccessServicesVersion** command is used to get the **Access Services Site Version** (section [3.1.1.2](#)) of the current site or subsite.

The following is the WSDL port type specification of the **GetAccessServicesVersion WSDL operation**.

```
<wsdl:operation name="GetAccessServicesVersion"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
```

```

    <wsdl:input message="tns:GetAccessServicesVersionSoapIn"/>
    <wsdl:output message="tns:GetAccessServicesVersionSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **GetAccessServicesVersionSoapIn** request message and the protocol server sends a **GetAccessServicesVersionSoapOut** response message as follows:

- In the event of an application error on the protocol server during this operation, a SOAP fault is returned, as described in section [2.2.2.1](#).
- If the **GetAccessServicesVersionSoapIn** request message is sent to an **Access Services** site, the protocol server MUST return the **Access Services Site Version** of that site. If the request was sent to a site that is not an Access Services site, the protocol server MUST return a major version of "-1" and a minor version of zero ("0").

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetAccessServicesVersionSoapIn	The request WSDL message for the GetAccessServicesVersion WSDL operation.
GetAccessServicesVersionSoapOut	The response WSDL message for the GetAccessServicesVersion WSDL operation.

3.1.4.1.1.1 GetAccessServicesVersionSoapIn

The request WSDL message for the **GetAccessServicesVersion** WSDL operation.

The **SOAP action** value is:

```

http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetAccessServicesVersion

```

The **SOAP body** contains the **GetAccessServicesVersion** element.

3.1.4.1.1.2 GetAccessServicesVersionSoapOut

The response WSDL message for the **GetAccessServicesVersion** WSDL operation.

The SOAP body contains the **GetAccessServicesVersionResponse** element.

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetAccessServicesVersion	The input data for the GetAccessServicesVersion WSDL operation.
GetAccessServicesVersionResponse	The result data for the GetAccessServicesVersion WSDL operation.

3.1.4.1.2.1 GetAccessServicesVersion

The **GetAccessServicesVersion** element specifies the input data for the **GetAccessServicesVersion** WSDL operation.

```
<xs:element name="GetAccessServicesVersion" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType/>
</xs:element>
```

3.1.4.1.2.2 GetAccessServicesVersionResponse

The **GetAccessServicesVersionResponse** element specifies the result data for the **GetAccessServicesVersion** WSDL operation.

```
<xs:element name="GetAccessServicesVersionResponse"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="Version" nillable="true"
type="tns:VersionType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Version: The **Access Services Site Version** of the site or subsite as specified in section [2.2.4.3](#).

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 GetCurrentUserInfo

The **GetCurrentUserInfo** operation returns XML that describes the user profile being used when sending requests to the protocol server.

The following is the WSDL port type specification of the **GetCurrentUserInfo** WSDL operation.

```
<wsdl:operation name="GetCurrentUserInfo" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetCurrentUserInfoSoapIn"/>
</wsdl:operation>
```

```
<wsdl:output message="tns:GetCurrentUserInfoSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetCurrentUserInfoSoapIn** request message and the protocol server responds with a **GetCurrentUserInfoSoapOut** response message as follows:

- In the event of an application error on the protocol server during this operation, a SOAP fault is returned as described in section [2.2.2.1](#).
- Otherwise, the protocol server MUST respond with a **GetCurrentUserInfoSoapOut** response message that contains information about the user profile.

3.1.4.2.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetCurrentUserInfoSoapIn	The request WSDL message for the GetCurrentUserInfo WSDL operation.
GetCurrentUserInfoSoapOut	The response WSDL message for the GetCurrentUserInfo WSDL operation.

3.1.4.2.1.1 GetCurrentUserInfoSoapIn

The request WSDL message for the **GetCurrentUserInfo** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetCurrentUserInfo
```

The SOAP body contains the **GetCurrentUserInfo** element.

3.1.4.2.1.2 GetCurrentUserInfoSoapOut

The response WSDL message for the **GetCurrentUserInfo** WSDL operation.

The SOAP body contains the **GetCurrentUserInfoResponse** element.

3.1.4.2.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetCurrentUserInfo	The input data for the GetCurrentUserInfo WSDL operation.
GetCurrentUserInfoResponse	The result data for the GetCurrentUserInfo WSDL operation.

3.1.4.2.2.1 GetCurrentUserInfo

The **GetCurrentUserInfo** element specifies the input data for the **GetCurrentUserInfo** WSDL operation.

```
<xs:element name="GetCurrentUserInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType/>
</xs:element>
```

3.1.4.2.2 GetCurrentUserInfoResponse

The **GetCurrentUserInfoResponse** element specifies the result data for the **GetCurrentUserInfo** WSDL operation.

```
<xs:element name="GetCurrentUserInfoResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetCurrentUserInfoResult">
        <xs:complexType mixed="true">
          <xs:sequence>
            <xs:any/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetCurrentUserInfoResult: This element MUST be present and MUST have one child element named **GetCurrentUserInfo**. The **GetCurrentUserInfo** element MUST be present and MUST have two child elements. The first MUST be an element named **User** of type **User**, as specified in [\[MS-UGS\]](#) section 2.2.4.7. The second element MUST be named **Groups** and be of type **Groups**, as specified in [\[MS-UGS\]](#) section 2.2.4.2.

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 GetDataMacroState

The **GetDataMacroState** command is used to get the current state of a data macro ([\[MS-AXL\]](#) section 2.1.3.2).

The following is the WSDL port type specification of the **GetDataMacroState** WSDL operation.

```

<wsdl:operation name="GetDataMacroState" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetDataMacroStateSoapIn"/>
  <wsdl:output message="tns:GetDataMacroStateSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **GetDataMacroStateSoapIn** request message and the protocol server sends a **GetDataMacroStateSoapOut** response message, as follows:

- If the specified **macroToken** element is a valid string that represents a data macro that was triggered on the protocol server, the protocol server MUST respond with the current state of the data macro.
- If the specified **macroToken** is not a valid string, or does not represent a data macro on the protocol server, a SOAP fault is returned as described in section [2.2.2.1](#).

3.1.4.3.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetDataMacroStateSoapIn	The request WSDL message for the GetDataMacroState WSDL operation.
GetDataMacroStateSoapOut	The response WSDL message for the GetDataMacroState WSDL operation.

3.1.4.3.1.1 GetDataMacroStateSoapIn

The request WSDL message for the **GetDataMacroState** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetDataMacroState
```

The SOAP body contains the **GetDataMacroState** element.

3.1.4.3.1.2 GetDataMacroStateSoapOut

The response WSDL message for the **GetDataMacroState** WSDL operation.

The SOAP body contains the **GetDataMacroStateResponse** element.

3.1.4.3.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetDataMacroState	The input data for the GetDataMacroState WSDL operation.
GetDataMacroStateResponse	The result data for the GetDataMacroState WSDL operation.

3.1.4.3.2.1 GetDataMacroState

The **GetDataMacroState** element specifies the input data for the **GetDataMacroState** WSDL operation.

```
<xs:element name="GetDataMacroState" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="macroToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

macroToken: The identifier of a data macro instance. This element **MUST** be present, and **MUST** have a value obtained from one of the following:

- The **RunDataMacroResult** element of a **RunDataMacroResponse** element returned by the **RunDataMacro** operation as specified in section [3.1.4.5.2.2](#).
- The **mit** attribute of an **Update** element in an **UpdateListsResponse** element returned by the **UpdateLists** operation as specified in section [3.1.4.8.2.2](#).

3.1.4.3.2.2 GetDataMacroStateResponse

The **GetDataMacroStateResponse** element specifies the result data for the **GetDataMacroState** WSDL operation.

```
<xs:element name="GetDataMacroStateResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetDataMacroStateResult"
type="tns:DataMacroInstanceState"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetDataMacroStateResult: The current state of the data macro instance as specified in section [3.1.4.3.3.1](#). **MUST** be specified.

3.1.4.3.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
DataMacroInstanceState	The state of a data macro instance on the server, including any errors and return variables, as specified in [MS-AXL] section 2.1.3.3.3.

3.1.4.3.3.1 DataMacroInstanceState

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

The state of a data macro instance on the server, including any errors and return variables, as specified in [\[MS-AXL\]](#) section 2.1.3.3.3.

```

<xs:complexType name="DataMacroInstanceState" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="State" type="tns:DataMacroState"/>
    <xs:element minOccurs="1" maxOccurs="1" name="ErrorNumber" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ErrorDescription" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ReturnVars"
type="tns:ArrayOfKeyValuePair"/>
  </xs:sequence>
</xs:complexType>

```

State: The current state of the data macro instance as specified in section [3.1.4.3.4.1](#).

ErrorNumber: The error number returned by the data macro. See [MS-AXL] section 2.2.5.1.14.

ErrorDescription: The error description returned by the data macro. See [MS-AXL] section 2.2.5.1.14.

ReturnVars: Specifies the return variables, as specified in [MS-AXL] section 2.1.3.3.3, of the data macro. This MUST be an **ArrayOfKeyValuePair**, as specified in section [2.2.4.1](#), that MUST contain a **KeyValuePair** for each return variable returned from the data macro, and the **Key** of the **KeyValuePair** MUST be the same as the name of the return variable it is representing.

3.1.4.3.4 Simple Types

The following table summarizes the XML schema simple type definitions that are specific to this operation.

Simple type	Description
DataMacroState	The current state of a data macro instance.

3.1.4.3.4.1 DataMacroState

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

The current state of a data macro instance.

```

<xs:simpleType name="DataMacroState" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Error"/>
    <xs:enumeration value="Running"/>
    <xs:enumeration value="Complete"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for the **DataMacroState** simple type.

Value	Meaning
Error	The current data macro has stopped running because of an error.
Running	The current data macro is still running.
Complete	The current data macro has successfully finished running with no errors.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 GetServerInformation

The **GetServerInformation** command is used to get information about which **Access Services Site Versions** (section [3.1.1.2](#)) the protocol server supports and the **Access Services Site Version** of the current site or subsite.

The following is the WSDL port type specification of the **GetServerInformation** WSDL operation.

```
<wsdl:operation name="GetServerInformation" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetServerInformationSoapIn"/>
  <wsdl:output message="tns:GetServerInformationSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetServerInformationSoapIn** request message, and the protocol server sends a **GetServerInformationSoapOut** response message, as follows:

- In the event of an application error on the protocol server during this operation, a SOAP fault is returned, as described in section [2.2.2.1](#).
- Otherwise, the protocol server MUST respond with a **GetServerInformationSoapOut** response message that contains information about the **Access Services Site Version** (section 3.1.1.2) supported on the protocol server. If the **GetServerInformationSoapIn** request message is sent to an **Access Services** site or subsite, the protocol server MUST also return the **Access Services Site Version** of that site or subsite. If the request was sent to a **top-level site** of a **site collection**, or a site that is not an **Access Services** site, the protocol server MUST return a major version of "-1" and a minor version of zero ("0").

3.1.4.4.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetServerInformationSoapIn	The request WSDL message for the GetServerInformation WSDL operation.
GetServerInformationSoapOut	The response WSDL message for the GetServerInformation WSDL operation.

3.1.4.4.1.1 GetServerInformationSoapIn

The request WSDL message for the **GetServerInformation** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetServerInformation
```

The SOAP body contains the **GetServerInformation** element.

3.1.4.4.1.2 GetServerInformationSoapOut

The response WSDL message for the **GetServerInformation** WSDL operation.

The SOAP body contains the **GetServerInformationResponse** element.

3.1.4.4.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetServerInformation	The input data for the GetServerInformation WSDL operation.
GetServerInformationResponse	The result data for the GetServerInformation WSDL operation.

3.1.4.4.2.1 GetServerInformation

The **GetServerInformation** element specifies the input data for the **GetServerInformation** WSDL operation.

```
<xs:element name="GetServerInformation" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType/>  
</xs:element>
```

3.1.4.4.2.2 GetServerInformationResponse

The **GetServerInformationResponse** element specifies the result data for the **GetServerInformation** WSDL operation.

```
<xs:element name="GetServerInformationResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="1" maxOccurs="1" name="AccessServerInformation" nillable="true"  
        type="tns:AccessServerInformationType"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

AccessServerInformation: Information about the minimum and maximum **Access Services Site Version**, as specified in section [3.1.1.2](#), supported by the protocol server, and the current **Access Services Site Version**, as specified in section [3.1.4.4.3.1](#).

3.1.4.4.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
AccessServerInformationType	The AccessServerInformationType complex type contains information about the minimum and maximum Access Services Site Version (section 3.1.1.2) the protocol server supports, and the Access Services Site Version of the current site or subsite.

3.1.4.4.3.1 AccessServerInformationType

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

The **AccessServerInformationType** complex type contains information about the minimum and maximum **Access Services Site Version** (section [3.1.1.2](#)) the protocol server supports, and the **Access Services Site Version** of the current site or subsite.

```
<xs:complexType name="AccessServerInformationType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="MinimumAccessServicesVersion"
type="tns:VersionType"/>
    <xs:element minOccurs="0" maxOccurs="1" name="MaximumAccessServicesVersion"
type="tns:VersionType"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SiteVersion" type="tns:VersionType"/>
  </xs:sequence>
</xs:complexType>
```

MinimumAccessServicesVersion: The minimum **Access Services Site Version** that is supported on the protocol server. The format of this element MUST be as specified in section [2.2.4.3](#).

MaximumAccessServicesVersion: The maximum **Access Services Site Version** that is supported on the protocol server. The format of this element MUST be as specified in section [2.2.4.3](#).

SiteVersion: The current **Access Services Site Version** of the site or subsite, as specified in section [2.2.4.3](#). If the request was sent to a top-level site of a site collection, this MUST have a major version of "-1" and a minor version of zero ("0").

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 RunDataMacro

The **RunDataMacro** operation triggers the running of a data macro ([\[MS-AXL\]](#) section 2.2.1.5).

The following is the WSDL port type specification of the **RunDataMacro** WSDL operation.

```
<wsdl:operation name="RunDataMacro" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:RunDataMacroSoapIn"/>
  <wsdl:output message="tns:RunDataMacroSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RunDataMacroSoapIn** request message and the protocol server responds with a **RunDataMacroSoapOut** response message as follows:

- In the event of an application error on the **protocol server** during this operation, a SOAP fault is returned as described in section [2.2.2.1](#).
- If the macro name specified by the protocol client does not match the name of a data macro on the protocol server, a SOAP fault is returned as described in section 2.2.2.1.
- Otherwise, the protocol server **MUST** respond with a **RunDataMacroSoapOut** response message that contains an identifier for the data macro instance the protocol server started.

3.1.4.5.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
RunDataMacroSoapIn	The request WSDL message for the RunDataMacro WSDL operation.
RunDataMacroSoapOut	The response WSDL message for the RunDataMacro WSDL operation.

3.1.4.5.1.1 RunDataMacroSoapIn

The request WSDL message for the **RunDataMacro** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/Webservices/AccessServer/RunDataMacro
```

The SOAP body contains the **RunDataMacro** element.

3.1.4.5.1.2 RunDataMacroSoapOut

The response WSDL message for the **RunDataMacro** WSDL operation.

The SOAP body contains the **RunDataMacroResponse** element.

3.1.4.5.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
RunDataMacro	The input data for the RunDataMacro WSDL operation.
RunDataMacroResponse	The result data for the RunDataMacro WSDL operation.

3.1.4.5.2.1 RunDataMacro

The **RunDataMacro** element specifies the input data for the **RunDataMacro** WSDL operation.

```
<xs:element name="RunDataMacro" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="macroName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:ArrayOfKeyValuePair"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

macroName: The name of the data macro to be run. MUST be specified. The value of **macroName** MUST be the title of a list within the current site, followed by a period, followed by the name of the data macro to run. The title of the list and the name of the data macro MUST be of type **ST_ObjectName**, as specified in [\[MS-AXL\]](#) section 2.2.4.1. If the title of the list contains a space, it MUST be surrounded by square brackets. If the name of the data macro contains a space, it MUST be surrounded by square brackets.

parameters: Specifies the parameters for the data macro. MUST be present if the data macro requires parameters, as specified in the **Parameters** element of a **CT_DataMacro**, as specified in ([\[MS-AXL\]](#) section 2.2.3.49. Comparisons between the **Key** in a **KeyValuePair** and the parameters in a data macro MUST be case insensitive. If a parameter is specified in more than one **KeyValuePair**, the protocol server MUST use the last **Value**. If any **Key** in **KeyValuePair** does not match a parameter in the data macro, it MUST be ignored.

3.1.4.5.2.2 RunDataMacroResponse

The **RunDataMacroResponse** element specifies the result data for the **RunDataMacro** WSDL operation.

```
<xs:element name="RunDataMacroResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="RunDataMacroResult" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

RunDataMacroResult: A new identifier for the data macro instance that was triggered. MUST be specified. This identifier can be supplied to **GetDataMacroState** (section [3.1.4.3](#)) to obtain state information about the data macro instance.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 SetAccessServicesVersion

The **SetAccessServicesVersion** command is used to set the **Access Services Site Version** (section [3.1.1.2](#)) of the current site or subsite.

The following is the WSDL port type specification of the **SetAccessServicesVersion** WSDL operation.

```
<wsdl:operation name="SetAccessServicesVersion"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetAccessServicesVersionSoapIn"/>
  <wsdl:output message="tns:SetAccessServicesVersionSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **SetAccessServicesVersionSoapIn** request message and the protocol server sends a **SetAccessServicesVersionSoapOut** response message as follows:

- In the event of an application error on the protocol server during this operation, a SOAP fault is returned as described in section [2.2.2.1](#).
- If the version specified by **Version** is not supported by the protocol server, or is an invalid **VersionType**, a SOAP fault is returned as described in section 2.2.2.1.
- Otherwise, if the protocol server is a site, set the **Access Services Site Version** (section 3.1.1.2) to the values specified by the **Version** element of the request. The protocol server MUST respond with a **SetAccessServicesVersionSoapOut** response message.

3.1.4.6.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
SetAccessServicesVersionSoapIn	The request WSDL message for the SetAccessServicesVersion WSDL operation.
SetAccessServicesVersionSoapOut	The response WSDL message for the SetAccessServicesVersion WSDL operation.

3.1.4.6.1.1 SetAccessServicesVersionSoapIn

The request WSDL message for the **SetAccessServicesVersion** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/SetAccessServicesVersion
```


The SOAP body contains the **SetAccessServicesVersion** element.

3.1.4.6.1.2 SetAccessServicesVersionSoapOut

The response WSDL message for the **SetAccessServicesVersion** WSDL operation.

The SOAP body contains the **SetAccessServicesVersionResponse** element.

3.1.4.6.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
SetAccessServicesVersion	The input data for the SetAccessServicesVersion WSDL operation.
SetAccessServicesVersionResponse	The result data for the SetAccessServicesVersion WSDL operation.

3.1.4.6.2.1 SetAccessServicesVersion

The **SetAccessServicesVersion** element specifies the input data for the **SetAccessServicesVersion** WSDL operation.

```
<xs:element name="SetAccessServicesVersion" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="Version" nillable="true"
type="tns:VersionType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Version: The version of the site or subsite as specified in section [2.2.4.3](#).

3.1.4.6.2.2 SetAccessServicesVersionResponse

The **SetAccessServicesVersionResponse** element specifies the result data for the **SetAccessServicesVersion** WSDL operation.

```
<xs:element name="SetAccessServicesVersionResponse"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType/>
</xs:element>
```

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 StartCompilation

The **StartCompilation** command is used to notify a protocol server to start compiling the database application defined in the MSysASO list (section [3.1.1.1.1](#)) of the current site or subsite from the object definitions stored within that list.

The following is the WSDL port type specification of the **StartCompilation** WSDL operation.

```
<wsdl:operation name="StartCompilation" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:StartCompilationSoapIn"/>
  <wsdl:output message="tns:StartCompilationSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **StartCompilationSoapIn** request message and the protocol server responds with a **StartCompilationSoapOut** response message as follows:

- In the event of an application error on the protocol server during this operation, a SOAP fault is returned as described in section [2.2.2.1](#).
- Otherwise, the protocol server MUST respond with a **StartCompilationSoapOut** response message.

3.1.4.7.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
StartCompilationSoapIn	The request WSDL message for the StartCompilation WSDL operation.
StartCompilationSoapOut	The response WSDL message for the StartCompilation WSDL operation.

3.1.4.7.1.1 StartCompilationSoapIn

The request WSDL message for the **StartCompilation** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/StartCompilation
```

The SOAP body contains the **StartCompilation** element.

3.1.4.7.1.2 StartCompilationSoapOut

The response WSDL message for the **StartCompilation** WSDL operation.

The SOAP body contains the **StartCompilationResponse** element.

3.1.4.7.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
StartCompilation	The input data for the StartCompilation WSDL operation.
StartCompilationResponse	The result data for the StartCompilation WSDL operation.

3.1.4.7.2.1 StartCompilation

The **StartCompilation** element specifies the input data for the **StartCompilation** WSDL operation.

```
<xs:element name="StartCompilation" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType/>  
</xs:element>
```

3.1.4.7.2.2 StartCompilationResponse

The **StartCompilationResponse** element specifies the result data for the **StartCompilation** WSDL operation.

```
<xs:element name="StartCompilationResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType/>  
</xs:element>
```

3.1.4.7.3 Complex Types

None.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 UpdateLists

The **UpdateLists** operation is used to insert, update, and delete list items in one or more lists.

The following is the WSDL port type specification of the **UpdateLists** WSDL operation.

```
<wsdl:operation name="UpdateLists" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:UpdateListsSoapIn"/>
  <wsdl:output message="tns:UpdateListsSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **UpdateListsSoapIn** request message and the protocol server sends an **UpdateListsSoapOut** response message that consists of zero or more **u** elements. The protocol server MUST process each **u** element as follows:

1. If the list name specified in the **In** attribute of the **u** element is a valid **GUID** and corresponds to the identifier of a list on the site, use that list for the operation.
2. If the specified list name is not a valid GUID or does not correspond to the identifier of a list on the site, check if the list name corresponds to the title of a list on the site. If it does, use that list if performing either an **Insert** or **Update** command. If the list name corresponds to the title of a list on the site and performing a **Delete** command, return an error code of "-2130575322" in the **ec** attribute of the **u** element in the response that corresponds to this part of the request.
3. If the specified list name does not match a list based on either of the two previous conditions, the protocol server MUST return a SOAP fault, as described in section [2.2.2.1](#).
4. If there is a macro token specified in the **mit** element of the request, the protocol server MUST wait for the data macro that corresponds to that macro token to finish running before processing the **u** elements. If the request times out in an implementation-specific period of time, the protocol server MUST return an error code of "-2147467259" in the **ec** attribute of the **u** elements that failed to be updated.
5. Otherwise, the protocol server MUST process the operations on the list and return success or failure in the **ec** attribute of the **u** element in the response that corresponds to this part of the request.

3.1.4.8.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
UpdateListsSoapIn	The request WSDL message for the UpdateLists WSDL operation.
UpdateListsSoapOut	The response WSDL message for the UpdateLists WSDL operation.

3.1.4.8.1.1 UpdateListsSoapIn

The request WSDL message for the **UpdateLists** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/UpdateLists
```

The SOAP body contains the **UpdateLists** element.

3.1.4.8.1.2 UpdateListsSoapOut

The response WSDL message for the **UpdateLists** WSDL operation.

The SOAP body contains the **UpdateListsResponse** element.

3.1.4.8.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
UpdateLists	The input data for the UpdateLists WSDL operation.
UpdateListsResponse	The result data for the UpdateLists WSDL operation.

3.1.4.8.2.1 UpdateLists

The **UpdateLists** element specifies the input data for the **UpdateLists** WSDL operation.

```
<xs:element name="UpdateLists" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="u" type="tns:Update"/>
      <xs:element minOccurs="1" maxOccurs="1" name="par" type="xs:boolean"/>
      <xs:element minOccurs="1" maxOccurs="1" name="mit" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

u: Specifies an operation to perform on a list item. **MUST** be specified.

The following table lists the three different operations that can be performed on a list item. For more information about constructing a **u** element, see section [3.1.4.8.3.1](#)

Value	Description
Insert (i)	<p>Add a new list item to the specified list. For each f element within the u element, the protocol server MUST set the value of the field specified by the n attribute of that f element to the value stored in the v attribute of that element or return an error code in the ec attribute of the u element in the protocol server's response to this update.</p> <p>The protocol server MUST ignore the value in the id attribute of the u element in the client's request, and MUST return the assigned list item identifier of this inserted list item in the id attribute of the update element in the protocol server's response to this update.</p>
Update (u)	<p>Update fields for a specific list item. The id attribute of the u element MUST be specified with a value that equals the list item identifier of a list item that already exists on the protocol server. For each f element within the u element, the protocol server MUST set the value of the field specified by the n attribute of that f element to the value stored in the v attribute of that element, or return an error code in the ec attribute of the u element in the protocol server's response to this update.</p>

Value	Description
Delete (d)	Delete a specified list item. The id attribute of the u element MUST be specified with a value that equals the list item identifier of a list item that MUST be deleted by the protocol server, or return an error code in the ec attribute of the u element in the protocol server's response to this update.

There are two types of fields specified by the **n** attribute of an **f** element of the **u** element that require special treatment:

- owshiddenversion field (2)
- **lookup fields**

The **owshiddenversion** field, as specified in [\[MS-WSSTS\]](#) section 2.4.2, is used for conflict detection as follows:

- Insert operations MUST NOT have a value specified for the **owshiddenversion** field by the protocol client.
- Update operations use the **owshiddenversion** field as follows:
 - When the **owshiddenversion** field is not specified by the protocol client, the protocol server MUST overwrite any changes in the list item or return an error code in the **ec** attribute of the **u** element in the protocol server's response to this update indicating why the update failed.
 - When the **owshiddenversion** value specified by the protocol client is equal to the **owshiddenversion** field's (2) value for the list item on the protocol server, the protocol server MUST update the list item or return an error code in the **ec** attribute of the **u** element in the protocol server's response to this update indicating why the update failed. The protocol server MUST NOT update this field with the value that the protocol client includes in the request.
 - When the **owshiddenversion** specified by the protocol client is different from the current value of the **owshiddenversion** field for the list item on the protocol server, the protocol server MUST return error code "-2130575305" and return all of the values of the fields currently stored on the protocol server using the **f** elements of the **u** element in the response that corresponds to the current update.
- Delete operations MUST have a value specified for the **owshiddenversion** field by the protocol client. Delete operations use the **owshiddenversion** field as follows:
 - When the **owshiddenversion** value specified by the protocol client is equal to the **owshiddenversion** field's (2) value for the list item on the protocol server, the protocol server MUST delete the list item or return an error code in the **ec** attribute of the **u** element in the protocol server's response to this update indicating why the delete failed.
 - When the **owshiddenversion** specified by the protocol client is different from the current value of the **owshiddenversion** field for the list item on the protocol server, the protocol server MUST return error code "-2130575339" and return all of the values of the fields currently stored on the protocol server using the **f** elements of the **u** element in the response that corresponds to the current update.

If the field is a lookup field, as specified in [\[MS-WSSTS\]](#) section 2.3.1, the value MUST match a list item identifier in the target list of the lookup, or the protocol server MUST perform identifier fix up on the list item as specified in the following paragraph. The value of a lookup field might refer to a list item that is being inserted in the same **UpdateListsSoapIn** request.

If a lookup field is referenced in an **f** element of a **u** element with **cmd** attribute value "i" in a request from the protocol client, and the value of that lookup field, as specified by the **v** attribute of the **f** element, is set to a value that does not correspond to a list item identifier in the target list of the lookup, the list item that is the target of the lookup field has not yet been added to the protocol server. In this case, the value of the lookup field MUST reference the **id** of a **u** element in the current **UpdateLists** request, or the current **UpdateLists** request MUST have the **par** element set to "true". If the value of the lookup field references the **id** of a **u** element with **cmd** attribute value "i" in the current **UpdateLists** request, the protocol server MUST process the other updates in the **UpdateLists** request before attempting to process this update. After processing the other updates and assigning list item identifiers to any list items inserted by the **UpdateLists** request, the protocol server MUST replace the value of the lookup field with the actual list item identifier that the protocol server assigned to the inserted list item in the current request that had the **id** that matched the value of the lookup field, and expressed as specified in [MS-WSSTS] section 2.4.1. If the target of the lookup is not included in a previous **u** element from the current **u** element with **cmd** attribute value "i" in the same protocol client request, the protocol server MUST return an error in the **ec** attribute of the **u** element in the protocol server's response to this update, and MUST not commit any data for the current **u** element unless the **par** element of this **UpdateLists** request is set to "true", as specified in the following paragraph.

par: The value of this parameter determines whether insert operations in this update are treated as regular inserts or partial inserts. If **par** is set to "true", all **u** elements with a **cmd** attribute of value "i" are treated as partial inserts. Partial inserts can be used when the target of a lookup in a new list item will be inserted in a future request. In a partial insert, the server MUST commit any valid data in the **u** element, and MUST attempt to fix values in lookup fields, as specified in the preceding paragraph. If the protocol server cannot set a lookup field's value, the protocol server behavior is as follows:

- If the lookup field is not required to have a value, the protocol server MUST return "-2130575159" in the **ec** attribute of the **u** element in the protocol server's response to this update and MUST commit any valid data in the **u** element. A value MUST NOT be committed for the lookup field of this list item.
- If the lookup field is required to have a value, the protocol server MUST return "-2130575163" in the **ec** attribute of the **u** element in the protocol server's response to this update and MUST NOT commit any data from the **u** element.

If **par** is set to "false", all **u** elements with a **cmd** attribute value "i" will be treated as regular inserts. If all data in the **u** element is valid after attempting to fix values in the lookup fields, as previously specified, the protocol server MUST commit those values in a new list item. If the protocol server cannot set a lookup field's value for a **u** element, it must return "-2130575159" in the **ec** attribute of the **u** element in the protocol server's response to this update and MUST NOT commit any data from the **u** element.

mit: A macro token representing a data macro instance. If a macro token is specified, the protocol server MUST wait for the data macro instance to complete before attempting the update, and the macro token MUST have come from an **UpdateListsResultInfo** or a **RunDataMacroResponse**.

3.1.4.8.2.2 UpdateListsResponse

The **UpdateListsResponse** element specifies the result data for the **UpdateLists** WSDL operation.

```
<xs:element name="UpdateListsResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="UpdateListsResult"
        type="tns:UpdateListsResultInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

UpdateListsResult: Specifies the results for the UpdateLists request sent by the protocol client.

The number of elements in **UpdateListsResult** MUST be equal to the number of **Update** elements in the **UpdateLists** element supplied to the operation. If any of the **Update** elements use the **ut** attribute, the matching **Update** element in the response MUST have the same value for the **ut** attribute. This allows the protocol client to match responses and requests.

If a field has its **Hidden** attribute set to "true", as specified in [MS-WSSFO3] section 2.2.7.3.3.2, and the field is not the **ID** field or the **owshiddenversion** field, as specified in [MS-WSSTS] section 2.4.2, the protocol server MUST NOT return these fields (2) to the client in any response. Fields (2) of this type will be referenced later in insert and update responses.

For an insert request, if the operation is successful, the protocol server MUST return all of the fields that were created on the protocol server for the new list item, except for the fields that are hidden as specified earlier. The protocol server MUST return a new unique list item identifier as the **id** attribute of the response, which the protocol client MUST use to reference the list item in subsequent update or delete requests. If the operation is successful, the protocol server MUST set **ec** to zero ("0") and **em** to an empty string. Otherwise, the protocol server MUST set **ec** to an error code and MUST set **em** to an error message. See section 3.1.4.8.3.1 for a list of error codes.

For an update request, if the operation is successful, the protocol server MUST return all of the fields that have changed on the server, except for the fields that are hidden as specified earlier. The protocol server MUST return the columns **owshiddenversion**, **Modified**, and **Editor**, as defined in [MS-WSSTS] section 2.4.2, even if these fields have not changed. If the operation is successful, the protocol server MUST set **ec** to zero ("0") and **em** to an empty string. Otherwise, the protocol MUST set **ec** to an error code and MUST set **em** to an error message. See section 3.1.4.8.3.1 for a list of error codes.

For a delete request, if the operation is successful, the protocol server MUST return a **u** element with no **f** elements, and MUST set **ec** to zero ("0") and **em** to an empty string. If the operation is not successful, the protocol MUST set **ec** to an appropriate error code and MUST set **em** to an error message. See section 3.1.4.8.3.1 for a list of error codes.

3.1.4.8.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
FieldValue	The FieldValue complex type contains the name and value of a field for a specific list item.
Update	The Update complex type contains information about an insert, update, or delete to a particular list item.
UpdateListsResultInfo	The results of an UpdateLists request.

3.1.4.8.3.1 Update

Namespace: http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/

The **Update** complex type contains information about an insert, update, or delete to a particular list item.

```
<xs:complexType name="Update" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
```



```

    <xs:element minOccurs="0" maxOccurs="unbounded" name="f" nillable="true"
type="tns:FieldValue"/>
  </xs:sequence>
  <xs:attribute name="ec" type="xs:int" use="required"/>
  <xs:attribute name="em" type="xs:string"/>
  <xs:attribute name="cmd" type="tns:UpdateCommand" use="required"/>
  <xs:attribute name="ut" type="xs:string"/>
  <xs:attribute name="ln" type="xs:string"/>
  <xs:attribute name="id" type="xs:int" use="required"/>
</xs:complexType>

```

f: The **FieldValue** elements to be updated or that have been updated. See section [3.1.4.8.3.2](#) for details.

ec: The error code returned for this **Update**. This MUST only be set in a protocol server response. If the operation described by the current **Update** succeeded, this MUST be set to zero ("0"). The following table specifies error codes that MUST be returned for particular scenarios.

Error code	Error meaning
-2130575166	Delete cannot occur because of a restrict delete relationship.
-2130575339	There is a data conflict for a Delete operation because the information that is on the protocol server and the protocol client has different owshiddenversion field values.
-2130575322	A Delete operation was attempted by specifying a list by name in the In attribute.
-2130575305	There is a data conflict for an update operation because the information that is on the protocol server and the protocol client have different owshiddenversion field values.
-2130575169	An attempt was made to set a field that is marked to not allow duplicate values as a duplicate of another list item.
-2130575163	A value for a required field was not specified in the update request.
-2130575159	Trying to set the value of a lookup field to an ID that does not exist in the parent table.
-2147467259	General failure.

em: The error message returned for this **Update**. This MUST only be set in a protocol server response.

cmd: Specifies which **UpdateCommand** is being executed in this **Update**. MUST be present. See section [2.2.5.1](#) for details.

ut: An update tag used by the protocol client to associate protocol client requests with protocol server responses. If the protocol client is sending multiple **Update** elements in an **UpdateLists** request, the protocol client MUST specify a monotonically incrementing integer for **ut**, and the protocol server MUST send a reply with the same **ut**. If the protocol client is sending a single **Update** in the **UpdateLists** request, the protocol client MUST set **ut** to zero ("0"), and the protocol server MUST send a response with **ut** set to zero ("0").

In: The title or **list identifier** of the list that is being updated. MUST be present.

id: The list item identifier for the list item being updated. MUST be present.

3.1.4.8.3.2 FieldValue

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

The **FieldValue** complex type contains the name and value of a field for a specific list item.

```
<xs:complexType name="FieldValue" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="n" type="xs:string"/>
  <xs:attribute name="v" type="xs:string"/>
</xs:complexType>
```

n: Specifies the field internal name of a field in the list.

v: Specifies the value of the field. If present, **MUST** be a string representation of a value that matches the data type of the field specified by the **n** attribute, and **MUST** be formatted in the data culture of the session.

3.1.4.8.3.3 UpdateListsResultInfo

Namespace: <http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/>

The results of an **UpdateLists** request.

```
<xs:complexType name="UpdateListsResultInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="mit" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Update" type="tns:Update"/>
  </xs:sequence>
</xs:complexType>
```

mit: If the protocol server has triggered an **AfterInsert**, **AfterUpdate**, or **AfterDelete** data macro, as specified in [\[MS-AXL\]](#) section 2.1.3.2, in response to this update, this attribute specifies a macro token that can be used as an input to **GetDataMacroState** or **UpdateLists** operations. If the protocol server does not trigger an **AfterInsert**, **AfterUpdate**, or **AfterDelete** data macro, **mit** **MUST** be NULL.

Update: An array of **Update**. Specifies the results of all of the updates that were executed as part of the **UpdateLists** request. See section [3.1.4.8.3.1](#).

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 GetCurrentUserInfo

This example describes how **GetCurrentUserInfo** method works. To get the current user information, the protocol client sends the following message to the protocol server:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrentUserInfo
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
    >
      </GetCurrentUserInfo>
    </soap:Body>
  </soap:Envelope>
```

The protocol server then responds with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrentUserInfoResponse
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
    >
      <GetCurrentUserInfoResult>
        <GetCurrentUserInfo
          xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
        >
          <User ID="25" Sid="S-1-5-21-2127521184-1604012920-1887927527-4150143"
            Name="Andrew Cencini" LoginName="Northwind\andrew"
            Email="andrew@northwindtraders.com"
            Notes=""
            IsSiteAdmin="True"
            IsDomainGroup="False" Flags="0" />
          <Groups>
            <Group ID="3" Name="Team Site Owners" Description="Use this group to give people
            full control permissions to the SharePoint site: Team Site"
              OwnerID="3" OwnerIsUser="False" />
          </Groups>
        </GetCurrentUserInfo>
      </GetCurrentUserInfoResult>
    </GetCurrentUserInfoResponse>
  </soap:Body>
</soap:Envelope>
```

4.2 Use UpdateLists to Insert Items into a List

This example describes how to use **UpdateLists** method to insert list items into a list.

The protocol client sends the following message to the protocol server to insert a list item into a list:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateLists
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
    >
      <u cmd="i" ln="{3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}" ut="0" id="0">
        <f n="JobTitle" v="Sales Representative" />
      </u>
      <par>false</par>
    </UpdateLists>
  </soap:Body>
```

```
</soap:Envelope>
```

In the **u** element, as defined in section [3.1.4.8.3.1](#), the **cmd** attribute has the value "i", indicating that this is an insertion.

ln equals "{3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}". In this example, this is the GUID of an existing list. The insertion happens to this list.

ut equals zero ("0"). This value is used to match the **UpdateLists** response with **UpdateLists** request. The **ut** value in the corresponding response for this particular **UpdateLists** request is also zero ("0").

In an **insert** command, the **id** attribute in the request is ignored on the protocol server. The protocol client in this example sends a zero ("0").

In this example, the protocol client requests to insert a list item that has one field, named "JobTitle". Therefore, there is an **f** element, as defined in section [2.2.4.1](#).

The protocol server responds with the following message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateListsResponse
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/">
      <UpdateListsResult>
        <mit xsi:nil="true" />
        <Update ec="0" em="" cmd="i" ut="0" ln="{3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}"
          id="1">
          <f n="JobTitle" v=" Sales Representative " />
          <f n="LinkTitleNoMenu" v="" />
          <f n="Editor" v="25;#Andrew Cencini" />
          <f n="Author" v="25;#Andrew Cencini " />
          <f n="Modified" v="05/11/2009 08:02:36" />
          <f n="Created" v="05/11/2009 08:02:36" />
          <f n="ID" v="5" />
          <f n="owshiddenversion" v="1" />
          <f n="Attachments" v="False" />
        </Update>
      </UpdateListsResult>
    </UpdateListsResponse>
  </soap:Body>
</soap:Envelope>
```

In this example, the **id** attribute equals "1", meaning that the list item inserted has an identifier equal to "1" in the list. In subsequent **UpdateLists** calls, both the protocol client and the protocol server use an identifier equal to "1" to refer to this list item.

4.3 Use UpdateLists to Insert Items into Two Lists with Lookup Relationships

This example shows how to use **UpdateLists** to insert list items into two lists that have lookup relationships. In this example, List 1 is referred to by GUID {E5BDB272-1DFB-4752-903E-BF7BFF2052FE}. List 2 is referred to by GUID {3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}. List 1 has a field named "Occupation" that is of lookup type. It references the **id** attribute of the list items in List 2. The list item to be inserted into List 1 references a list item that is to be inserted into List 2 in the same **UpdateLists** call.

The protocol client sends the following message to the protocol server to insert list items into two lists in one call:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateLists
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/">
      <u cmd="i" ln="{3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}" ut="0" id="-1">
        <f n="JobTitle" v="Sales Representative" />
        <f n="_OldID" v="2" />
      </u>
      <u cmd="i" ln="{E5BDB272-1DFB-4752-903E-BF7BFF2052FE}" ut="4" id="-1">
        <f n="FullName" v="Nancy Freehafer" />
        <f n="Account" v="nancy@northwindtraders.com" />
        <f n="_OldID" v="4" />
        <f n="Occupation" v="-1" />
      </u>
      <par>false</par>
    </UpdateLists>
  </soap:Body>
</soap:Envelope>

```

Because there are two lists to be updated, there are two **u** elements in the message that is shown later. In the first **u** element, **ln** equals {3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}, meaning the insertion happens to List 2. The second **u** element in the message that is shown later has the **ln** attribute equals {E5BDB272-1DFB-4752-903E-BF7BFF2052FE}, referring to List 1.

If the list item to be inserted to List 1 references a list item that already exists in List 2 on the protocol server, the value of the field named "Occupation" is filled with the identifier of the list item being referenced. However, in this example, the list item to be inserted to List 1 references a list item that is to be inserted in List 2 in the same **UpdateLists** call. Therefore, at the time the protocol client sends the message, it does not know the identifier value for the list item being referenced. However, it knows that in the first **u** element, the **id** attribute equals "-1". That is the list item to be inserted into List 2. Therefore, the value of the fourth **f** element in the second **u** element is "-1" to refer to the list item to be inserted in List 2 in the first **u** element of this **UpdateLists** call.

The protocol server responds with the following message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateListsResponse
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/">
      <UpdateListsResult>
        <mit xsi:nil="true" />
        <Update ec="0" em="" cmd="i" ut="0" ln="{3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}"
          id="1" >
          <f n="JobTitle" v="Sales Representative" />
          <f n="_OldID" v="2" />
          <f n="LinkTitleNoMenu" v="" />
          <f n="Editor" v="25;#Andrew Cencini" />
          <f n="Author" v="25;#Andrew Cencini" />
          <f n="Modified" v="05/7/2009 06:32:01" />
          <f n="Created" v="05/7/2009 06:32:01"/>
          <f n="ID" v="1" />
          <f n="owshiddenversion" v="1" />
          <f n="Attachments" v="False" />
        </Update>
        <Update ec="0" em="" cmd="i" ut="4" ln="{E5BDB272-1DFB-4752-903E-BF7BFF2052FE}"
          id="1" >
          <f n="FullName" v="Nancy Freehafer" />
          <f n="Account" v="nancy@northwindtraders.com" />
          <f n="_OldID" v="4" />
          <f n="LinkTitleNoMenu" v="" />
          <f n="Editor" v="25;#Andrew Cencini" />
          <f n="Author" v="25;#Andrew Cencini" />
        </Update>
      </UpdateListsResult>
    </UpdateListsResponse>
  </soap:Body>
</soap:Envelope>

```

```

    <f n="Modified" v="05/7/2009 06:32:01" />
    <f n="Created" v="05/7/2009 06:32:01" />
    <f n="Occupation" v="1;#Sales Representative" />
    <f n="ID" v="1" />
    <f n="owshiddenversion" v="1" />
    <f n="Attachments" v="False" />
  </Update>
</UpdateListsResult>
</UpdateListsResponse>
</soap:Body>
</soap:Envelope>

```

In the first **Update** element, the list item is assigned an **id** attribute value "1" in List 2 ({3B6DEE82-D5AC-4ACE-A6E1-00774FA1E10F}). Once the new list item was created and assigned a permanent **id**, the protocol server performed identifier fix up on the remaining list items. Therefore, in the second **Update** element, the **f** element named "Occupation" in List 1 ({E5BDB272-1DFB-4752-903E-BF7BFF2052FE}) has the value "1; # Sales Representative" because it is referring to the list item with **id** = 1 in List 2.

4.4 RunDataMacro and GetDataMacroState

The following example describes how to use the **RunDataMacro** method to trigger a data macro called **AddAComment** and how to use the **GetDataMacroState** method to track the status of this data macro instance. In this example, the data macro **AddAComment** has two parameters, **ContactIDParam** and **CommentParam**. The protocol client sends the following message to the protocol server.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RunDataMacro
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/">
      <macroName>Comments.AddAComment</macroName>
      <parameters>
        <KeyValuePair>
          <Key xsi:type="xsd:string">contactIDParam</Key>
          <Value xsi:type="xsd:double">1</Value>
        </KeyValuePair>
        <KeyValuePair>
          <Key xsi:type="xsd:string">CommentParam</Key>
          <Value xsi:type="xsd:string"><![CDATA[abcded]]></Value>
        </KeyValuePair>
      </parameters>
    </RunDataMacro>
  </soap:Body>
</soap:Envelope>

```

The protocol server responds with the following message, which contains the token for this data macro instance on the protocol server.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RunDataMacroResponse
      xmlns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/">
      <RunDataMacroResult>AAEAAAD/////AQAAAAAAAAAAMAgAAAGJNaWNyb3NvZ
      nQuT2ZmaWNlLkFjY2Vzcy5TZXJ2ZXIsIFZlcnNpb249MTQuMC4LjAsIEN1bHR1cmU9bm
      VldHJhbCwgUHViGljS2V5VG9rZW49NzFlOWJjZTEwMjU5NDI5YwUBAAANU1pY3Jvc2
      9mdC5PZmZpY2UuQWNjZXNzLlNlcnZlci5EYXRhLkRhZGFNYWY0luc3RhbmNlBAAAAA
      xtX2luc3RhbmNlSWQNbV9zZXNzaW9uTmFtZRRtX3dvcmtmbG93SW5zdGFuY2VJZAdtX3
      N0YXRlAQEBBDJNaWNyb3NvZnQuT2ZmaWNlLkFjY2Vzcy5TZXJ2ZXIuRGF0YS5EYXRhTW

```



```
        <ErrorNumber>0</ErrorNumber>
      </GetDataMacroStateResult>
    </GetDataMacroStateResponse>
  </soap:Body>
</soap:Envelope>
```

In the response from the protocol server, the **GetDataMacroStateResult** element has the value "Running", meaning that the data macro is still processing.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:tns="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/"
>
      <xs:element name="UpdateLists">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="u" type="tns:Update"/>
            <xs:element minOccurs="1" maxOccurs="1" name="par" type="xs:boolean"/>
            <xs:element minOccurs="1" maxOccurs="1" name="mit" nillable="true"
type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Update">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="f" nillable="true"
type="tns:FieldValue"/>
        </xs:sequence>
        <xs:attribute name="ec" type="xs:int" use="required"/>
        <xs:attribute name="em" type="xs:string"/>
        <xs:attribute name="cmd" type="tns:UpdateCommand" use="required"/>
        <xs:attribute name="ut" type="xs:string"/>
        <xs:attribute name="ln" type="xs:string"/>
        <xs:attribute name="id" type="xs:int" use="required"/>
      </xs:complexType>
      <xs:complexType name="FieldValue">
        <xs:attribute name="n" type="xs:string"/>
        <xs:attribute name="v" type="xs:string"/>
      </xs:complexType>
      <xs:simpleType name="UpdateCommand">
        <xs:restriction base="xs:string">
          <xs:enumeration value="u"/>
          <xs:enumeration value="i"/>
          <xs:enumeration value="d"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:element name="UpdateListsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="UpdateListsResult"
type="tns:UpdateListsResultInfo"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="UpdateListsResultInfo">
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" name="mit" nillable="true"
type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Update" type="tns:Update"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="GetDataMacroState">
        <xs:complexType>
          <xs:sequence>
```

```

        <xs:element minOccurs="0" maxOccurs="1" name="macroToken" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetDataMacroStateResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetDataMacroStateResult"
type="tns:DataMacroInstanceState"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="DataMacroInstanceState">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="State" type="tns:DataMacroState"/>
        <xs:element minOccurs="1" maxOccurs="1" name="ErrorNumber" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ErrorDescription" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ReturnVars"
type="tns:ArrayOfKeyValuePair"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="DataMacroState">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Error"/>
        <xs:enumeration value="Running"/>
        <xs:enumeration value="Complete"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="ArrayOfKeyValuePair">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="KeyValuePair"
type="tns:KeyValuePair"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="KeyValuePair">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Key"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Value" nillable="true"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="RunDataMacro">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="macroName" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:ArrayOfKeyValuePair"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="RunDataMacroResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="RunDataMacroResult"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCurrentUserInfo">
    <xs:complexType/>
</xs:element>
<xs:element name="GetCurrentUserInfoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetCurrentUserInfoResult">
                <xs:complexType mixed="true">
                    <xs:sequence>
                        <xs:any/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetServerInformation">
    <xs:complexType/>
</xs:element>
<xs:complexType name="AccessServerInformationType">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="MinimumAccessServicesVersion"
type="tns:VersionType"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MaximumAccessServicesVersion"
type="tns:VersionType"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SiteVersion" type="tns:VersionType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="VersionType">
    <xs:attribute name="Major" type="xs:int" use="required"/>
    <xs:attribute name="Minor" type="xs:int" use="required"/>
</xs:complexType>
<xs:element name="GetServerInformationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="AccessServerInformation"
nillable="true" type="tns:AccessServerInformationType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAccessServicesVersion">
    <xs:complexType/>
</xs:element>
<xs:element name="GetAccessServicesVersionResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="Version" nillable="true"
type="tns:VersionType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAccessServicesVersion">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="Version" nillable="true"
type="tns:VersionType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAccessServicesVersionResponse">
    <xs:complexType/>
</xs:element>
<xs:element name="StartCompilation">
    <xs:complexType/>
</xs:element>
<xs:element name="StartCompilationResponse">
    <xs:complexType/>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:portType name="AccessServerSoap">
    <wsdl:operation name="UpdateLists">
        <wsdl:input message="tns:UpdateListsSoapIn"/>
        <wsdl:output message="tns:UpdateListsSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetDataMacroState">
        <wsdl:input message="tns:GetDataMacroStateSoapIn"/>
        <wsdl:output message="tns:GetDataMacroStateSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="RunDataMacro">
        <wsdl:input message="tns:RunDataMacroSoapIn"/>
    </wsdl:operation>
</wsdl:portType>
</wsdl:binding>
</wsdl:service>
</wsdl:definitions>

```

```

        <wsdl:output message="tns:RunDataMacroSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetCurrentUserInfo">
        <wsdl:input message="tns:GetCurrentUserInfoSoapIn"/>
        <wsdl:output message="tns:GetCurrentUserInfoSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetServerInformation">
        <wsdl:input message="tns:GetServerInformationSoapIn"/>
        <wsdl:output message="tns:GetServerInformationSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetAccessServicesVersion">
        <wsdl:input message="tns:GetAccessServicesVersionSoapIn"/>
        <wsdl:output message="tns:GetAccessServicesVersionSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="SetAccessServicesVersion">
        <wsdl:input message="tns:SetAccessServicesVersionSoapIn"/>
        <wsdl:output message="tns:SetAccessServicesVersionSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="StartCompilation">
        <wsdl:input message="tns:StartCompilationSoapIn"/>
        <wsdl:output message="tns:StartCompilationSoapOut"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="AccessServerSoap" type="tns:AccessServerSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="UpdateLists">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/Update
Lists" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetDataMacroState">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetDat
aMacroState" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RunDataMacro">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/RunDat
aMacro" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCurrentUserInfo">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetCur
rentUserInfo" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

    <wsdl:operation name="GetServerInformation">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetServerInformation" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAccessServicesVersion">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetAccessServicesVersion" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAccessServicesVersion">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/SetAccessServicesVersion" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="StartCompilation">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/StartCompilation" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="AccessServerSoap12" type="tns:AccessServerSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="UpdateLists">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/UpdateLists" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetDataMacroState">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetDataMacroState" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

```

```

    <wsdl:operation name="RunDataMacro">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/RunDat
aMacro" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCurrentUserInfo">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetCur
rentUserInfo" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetServerInformation">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetSer
verInformation" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAccessServicesVersion">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/GetAcc
essServicesVersion" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAccessServicesVersion">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/SetAcc
essServicesVersion" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="StartCompilation">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/Server/WebServices/AccessServer/StartC
ompilation" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:message name="GetAccessServicesVersionSoapIn">
    <wsdl:part name="parameters" element="tns:GetAccessServicesVersion"/>
  </wsdl:message>

```



```

</wsdl:message>
<wsdl:message name="GetAccessServicesVersionSoapOut">
  <wsdl:part name="parameters" element="tns:GetAccessServicesVersionResponse"/>
</wsdl:message>
<wsdl:message name="GetCurrentUserInfoSoapIn">
  <wsdl:part name="parameters" element="tns:GetCurrentUserInfo"/>
</wsdl:message>
<wsdl:message name="GetCurrentUserInfoSoapOut">
  <wsdl:part name="parameters" element="tns:GetCurrentUserInfoResponse"/>
</wsdl:message>
<wsdl:message name="GetDataMacroStateSoapIn">
  <wsdl:part name="parameters" element="tns:GetDataMacroState"/>
</wsdl:message>
<wsdl:message name="GetDataMacroStateSoapOut">
  <wsdl:part name="parameters" element="tns:GetDataMacroStateResponse"/>
</wsdl:message>
<wsdl:message name="GetServerInformationSoapIn">
  <wsdl:part name="parameters" element="tns:GetServerInformation"/>
</wsdl:message>
<wsdl:message name="GetServerInformationSoapOut">
  <wsdl:part name="parameters" element="tns:GetServerInformationResponse"/>
</wsdl:message>
<wsdl:message name="RunDataMacroSoapIn">
  <wsdl:part name="parameters" element="tns:RunDataMacro"/>
</wsdl:message>
<wsdl:message name="RunDataMacroSoapOut">
  <wsdl:part name="parameters" element="tns:RunDataMacroResponse"/>
</wsdl:message>
<wsdl:message name="SetAccessServicesVersionSoapIn">
  <wsdl:part name="parameters" element="tns:SetAccessServicesVersion"/>
</wsdl:message>
<wsdl:message name="SetAccessServicesVersionSoapOut">
  <wsdl:part name="parameters" element="tns:SetAccessServicesVersionResponse"/>
</wsdl:message>
<wsdl:message name="StartCompilationSoapIn">
  <wsdl:part name="parameters" element="tns:StartCompilation"/>
</wsdl:message>
<wsdl:message name="StartCompilationSoapOut">
  <wsdl:part name="parameters" element="tns:StartCompilationResponse"/>
</wsdl:message>
<wsdl:message name="UpdateListsSoapIn">
  <wsdl:part name="parameters" element="tns:UpdateLists"/>
</wsdl:message>
<wsdl:message name="UpdateListsSoapOut">
  <wsdl:part name="parameters" element="tns:UpdateListsResponse"/>
</wsdl:message>
</wsdl:definitions>

```

7 Appendix B: Product Behavior

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

- Microsoft Access 2010
- Microsoft SharePoint Server 2010
- Microsoft Access 2013
- Microsoft SharePoint Server 2013
- Microsoft Access 2016
- 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.

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.
7 Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

9 Index

A

Abstract data model
[server](#) 18
 [Access services site template](#) 18
 [Access services site version](#) 26
[Access services site template](#) 18
 [MSysASO](#) 19
 [USysApplicationLob](#) 25
[Access services site version](#) 26
[Applicability](#) 12
[ArrayOfKeyValuePair complex type](#) 15
[Attribute groups](#) 17
[Attributes](#) 17

C

[Capability negotiation](#) 13
[Change tracking](#) 67
[Complex types](#) 15
 [ArrayOfKeyValuePair](#) 15
 [KeyValuePair](#) 15
 [VersionType](#) 16

D

Data model - abstract
[server](#) 18
 [Access services site template](#) 18
 [Access services site version](#) 26

E

Events
 [local - server](#) 51
 [timer - server](#) 51
Examples
 [GetCurrentUserInfo](#) 52
 [RunDataMacro and GetDataMacroStatus](#) 55
 [use UpdateLists to insert items into a list](#) 52
 [use UpdateLists to insert items into two lists with lookup relationships](#) 53

F

[Fields - vendor-extensible](#) 13
[Full WSDL](#) 59

G

[GetAccessServicesVersion operation](#) 27
[GetCurrentUserInfo example](#) 52
[GetCurrentUserInfo operation](#) 30
[GetDataMacroState operation](#) 32
[GetServerInformation operation](#) 35
[Glossary](#) 8
[Groups](#) 17

I

[Implementer - security considerations](#) 58

[Index of security parameters](#) 58
[Informative references](#) 11
Initialization
 [server](#) 27
[Introduction](#) 8

K

[KeyValuePair complex type](#) 15

L

Local events
 [server](#) 51

M

Message processing
 [server](#) 27
Messages
 [ArrayOfKeyValuePair complex type](#) 15
 [attribute groups](#) 17
 [attributes](#) 17
 [complex types](#) 15
 [elements](#) 15
 [enumerated](#) 15
 [Faults](#) 15
 [Faults message](#) 15
 [groups](#) 17
 [KeyValuePair complex type](#) 15
 [namespaces](#) 14
 [simple types](#) 16
 [syntax](#) 14
 [transport](#) 14
 [UpdateCommand simple type](#) 16
 [VersionType complex type](#) 16
MSysASO
 [Attachments](#) 24
 [ClientObject](#) 21
 [ClientObjectProperties](#) 24
 [Flags](#) 24
 [ID](#) 19
 [owshiddenversion](#) 19
 [Revision](#) 21
 [ServerObject](#) 23
 [Title](#) 19
 [Type](#) 19

N

[Namespaces](#) 14
[Normative references](#) 10

O

Operations
 [GetAccessServicesVersion](#) 27
 [GetCurrentUserInfo](#) 30
 [GetDataMacroState](#) 32
 [GetServerInformation](#) 35
 [RunDataMacro](#) 38

[SetAccessServicesVersion](#) 40
[StartCompilation](#) 42
[UpdateLists](#) 44
[Overview \(synopsis\)](#) 11

P

[Parameters - security index](#) 58
[Preconditions](#) 12
[Prerequisites](#) 12
[Product behavior](#) 66
Protocol Details
[overview](#) 18

R

[References](#) 10
[informative](#) 11
[normative](#) 10
[Relationship to other protocols](#) 12
[RunDataMacro and GetDataMacroStatus example](#) 55
[RunDataMacro operation](#) 38

S

Security
[implementer considerations](#) 58
[parameter index](#) 58
Sequencing rules
[server](#) 27
Server
[abstract data model](#) 18
[Access services site template](#) 18
[Access services site version](#) 26
[GetAccessServicesVersion operation](#) 27
[GetCurrentUserInfo operation](#) 30
[GetDataMacroState operation](#) 32
[GetServerInformation operation](#) 35
[initialization](#) 27
[local events](#) 51
[message processing](#) 27
[RunDataMacro operation](#) 38
[sequencing rules](#) 27
[SetAccessServicesVersion operation](#) 40
[StartCompilation operation](#) 42
[timer events](#) 51
[timers](#) 27
[UpdateLists operation](#) 44
[SetAccessServicesVersion operation](#) 40
[Simple types](#) 16
[UpdateCommand](#) 16
[Standards assignments](#) 13
[StartCompilation operation](#) 42
Syntax
[messages - overview](#) 14

T

Timer events
[server](#) 51
Timers
[server](#) 27
[Tracking changes](#) 67
[Transport](#) 14
Types

[complex](#) 15
[simple](#) 16

U

[UpdateCommand simple type](#) 16
[UpdateLists operation](#) 44
[Use UpdateLists to insert items into a list example](#) 52
[Use UpdateLists to insert items into two lists with lookup relationships example](#) 53
USysApplicationLob
[Category](#) 26
[Created](#) 25
[ID](#) 25
[owshiddenversion](#) 26

V

[Vendor-extensible fields](#) 13
[Versioning](#) 13
[VersionType complex type](#) 16

W

[WSDL](#) 59