

[MS-PEOPS]:

People Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might 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, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
7/13/2009	1.02	Major	Revised and edited the technical content
8/28/2009	1.03	Editorial	Revised and edited the technical content
11/6/2009	1.04	Editorial	Revised and edited the technical content
2/19/2010	2.0	Editorial	Revised and edited the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.5	Minor	Clarified the meaning of the technical content.
1/20/2012	2.6	Minor	Clarified the meaning of the technical content.
4/11/2012	2.6	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.6	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.6	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.7	Minor	Clarified the meaning of the technical content.
2/11/2013	3.0	Major	Significantly changed the technical content.
7/30/2013	3.1	Minor	Clarified the meaning of the technical content.
11/18/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.1	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
4/30/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	4.0	Major	Significantly changed the technical content.
2/26/2016	5.0	Major	Significantly changed the technical content.
7/15/2016	5.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	5.0	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	6.0	Major	Significantly changed the technical content.
10/1/2018	7.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	7
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement	8
1.7	Versioning and Capability Negotiation	8
1.8	Vendor-Extensible Fields	8
1.9	Standards Assignments.....	9
2	Messages.....	10
2.1	Transport	10
2.2	Common Message Syntax	10
2.2.1	Namespaces	10
2.2.2	Messages.....	10
2.2.3	Elements	11
2.2.4	Complex Types.....	11
2.2.4.1	ArrayOfPrincipalInfo	11
2.2.4.2	PrincipalInfo	11
2.2.5	Simple Types	12
2.2.5.1	SPPrincipalType	12
2.2.6	Attributes	13
2.2.7	Groups	13
2.2.8	Attribute Groups.....	13
3	Protocol Details.....	14
3.1	Protocol Server Details	14
3.1.1	Abstract Data Model.....	14
3.1.2	Timers	14
3.1.3	Initialization.....	14
3.1.4	Message Processing Events and Sequencing Rules	14
3.1.4.1	ResolvePrincipals	15
3.1.4.1.1	Messages	15
3.1.4.1.1.1	ResolvePrincipalsSoapIn	16
3.1.4.1.1.2	ResolvePrincipalsSoapOut	16
3.1.4.1.2	Elements.....	16
3.1.4.1.2.1	ResolvePrincipals	16
3.1.4.1.2.2	ResolvePrincipalsResponse	17
3.1.4.1.3	Complex Types	18
3.1.4.1.3.1	ArrayOfString	18
3.1.4.1.4	Simple Types	18
3.1.4.1.5	Attributes	18
3.1.4.1.6	Groups.....	18
3.1.4.1.7	Attribute Groups.....	18
3.1.4.2	SearchPrincipals	18
3.1.4.2.1	Messages	19
3.1.4.2.1.1	SearchPrincipalsSoapIn	19
3.1.4.2.1.2	SearchPrincipalsSoapOut	19
3.1.4.2.2	Elements.....	19
3.1.4.2.2.1	SearchPrincipals	20
3.1.4.2.2.2	SearchPrincipalsResponse	20
3.1.4.2.3	Complex Types	21

3.1.4.2.4	Simple Types	21
3.1.4.2.5	Attributes	21
3.1.4.2.6	Groups	21
3.1.4.2.7	Attribute Groups.....	21
3.1.4.3	IsClaimsMode	21
3.1.4.3.1	Messages	21
3.1.4.3.1.1	IsClaimsModeSoapIn	22
3.1.4.3.1.2	IsClaimsModeSoapOut	22
3.1.4.3.2	Elements.....	22
3.1.4.3.2.1	IsClaimsMode	22
3.1.4.3.2.2	IsClaimsModeResponse.....	22
3.1.4.3.3	Complex Types	23
3.1.4.3.4	Simple Types	23
3.1.4.3.5	Attributes	23
3.1.4.3.6	Groups.....	23
3.1.4.3.7	Attribute Groups.....	23
3.1.5	Timer Events.....	23
3.1.6	Other Local Events.....	23
4	Protocol Examples	24
4.1	Resolve Principals Knowing Their E-mail Addresses	24
4.2	Search for Principals Knowing the E-mail Address	25
4.3	Check Whether in Claims Mode	26
5	Security	27
5.1	Security Considerations for Implementers	27
5.2	Index of Security Parameters	27
6	Appendix A: Full WSDL	28
7	Appendix B: Product Behavior	32
8	Change Tracking.....	33
9	Index.....	34

1 Introduction

The People Web Service Protocol enables protocol clients to find and resolve authenticated entities on a protocol server. It allows the client to pass criteria consisting of a list of keys and the principal type to the protocol server and to receive back a list of resolved principals.

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:

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

forms authentication: An authentication method in which protocol clients redirect unauthenticated requests to an HTML form by using HTTP. If the protocol client authenticates the request, the system issues a cookie that stores the credentials or a key for reacquiring the identity. In subsequent requests, the cookie is submitted in request headers and the requests are authenticated and authorized by an ASP.NET event handler that uses the validation method that is specified by the protocol client.

group: A named collection of users who share similar access permissions or roles.

principal: An authenticated entity that initiates a message or channel in a distributed system.

security group: A named group of principals on a SharePoint site.

Session Initiation Protocol (SIP) address: A URI that does not include a "sip:" prefix and is used to establish multimedia communications sessions between two or more users over an IP network, as described in [\[RFC3261\]](#).

site: (1) A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as web site.

(2) 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 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.

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

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 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.

[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", W3C Note, 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/>

1.2.2 Informative References

None.

1.3 Overview

This protocol enables a protocol client to resolve and find **principals**. The protocol allows the protocol client to pass criteria, which consist of a list of keys and the principal (1) type, to the protocol server and to receive back a list of resolved principals (1).

A typical scenario for using this protocol is a people-finder application that enables users to choose the principal (1) type and to enter the user name, e-mail address, or display name as search criteria. The application could use this protocol to resolve and find the principals (1) and show details about them.

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:

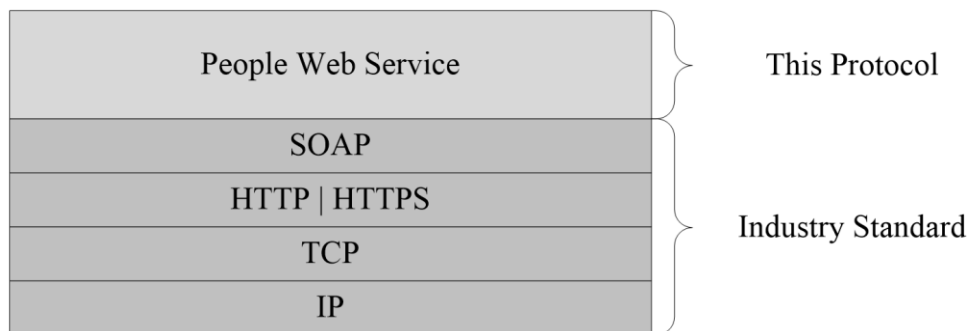


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **site (1)** that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending `"/_vti_bin/People.asmx"` to the URL of the site (1), for example: `http://www.contoso.com/Repository/_vti_bin/People.asmx`.

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

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported transports:** This protocol uses multiple transports with SOAP, as described in section [2.1](#).

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. Protocol servers **SHOULD** additionally support SOAP over HTTPS for securing communication with protocol clients.

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

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [\[XMLSCHEMA1/2\]](#) and [\[XMLSCHEMA2/2\]](#), and WSDL, as specified in [\[WSDL\]](#).

2.2.1 Namespaces

This protocol specifies and references **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this document associates an XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific. The following table describes these namespaces.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/sharepoint/soap/	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
(none)	http://schemas.microsoft.com/sharepoint/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

This specification does not define any common **XML schema** element definitions.

2.2.4 Complex Types

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

Complex type	Description
ArrayOfPrincipalInfo	Array of elements of type PrincipalInfo (section 2.2.4.2).
PrincipalInfo	Contains information about a principal (1).

2.2.4.1 ArrayOfPrincipalInfo

The **ArrayOfPrincipalInfo** complex type contains an array of **PrincipalInfo** elements (section [2.2.4.2](#)). This array contains zero or more **PrincipalInfo** elements (section 2.2.4.2).

```
<s:complexType name="ArrayOfPrincipalInfo">
  <s:sequence>
    <s:element name="PrincipalInfo" type="tns:PrincipalInfo" minOccurs="0"
maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

PrincipalInfo: The information about the principal (1) returned by the protocol server.

2.2.4.2 PrincipalInfo

The **PrincipalInfo** complex type provides information about a principal (1).

```
<s:complexType name="PrincipalInfo">
  <s:sequence>
    <s:element name="AccountName" type="s:string" minOccurs="0" maxOccurs="1" />
    <s:element name="UserInfoID" type="s:int" minOccurs="1" maxOccurs="1"/>
    <s:element name="DisplayName" type="s:string" minOccurs="0" maxOccurs="1" />
    <s:element name="Email" type="s:string" minOccurs="0" maxOccurs="1" />
    <s:element name="Department" type="s:string" minOccurs="0" maxOccurs="1" />
    <s:element name="Title" type="s:string" minOccurs="0" maxOccurs="1" />
    <s:element name="IsResolved" type="s:boolean" minOccurs="1" maxOccurs="1" />
    <s:element name="MoreMatches" type="tns:ArrayOfPrincipalInfo" minOccurs="0" maxOccurs="1"
  />
  <s:element name="PrincipalType" type="tns:SPPrincipalType" minOccurs="1" maxOccurs="1" />
  </s:sequence>
</s:complexType>
```

AccountName: The logon name of the principal (1).[<1>](#)

UserInfoID: The identifier of the user in the **site collection**.

Display Name: The display name of the principal (1).

Email: The e-mail address of the principal (1).

Department: The department name of the principal (1).

Title: The job title of the principal (1).

IsResolved: A Boolean that specifies whether the principal (1) is uniquely identified in the site collection list of users.

MoreMatches: The list of principals (1) that have name, display name, e-mail address, or **Session Initiation Protocol (SIP) address** starts with the input criteria. The list can be empty if there are no partially matching principals (1).

PrincipalType: The type of the principal (1). This value MUST be a **SPPrincipalType** (section [2.2.5.1](#)).

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
SPPrincipalType	Specifies the principal (1) type.

2.2.5.1 SPPrincipalType

The **SPPrincipalType** simple type is an enumeration that specifies the type of a principal (1).

```
<s:simpleType name="SPPrincipalType">
  <s:list>
    <s:simpleType>
      <s:restriction base="s:string">
        <s:enumeration value="None"/>
        <s:enumeration value="User"/>
        <s:enumeration value="DistributionList"/>
        <s:enumeration value="SecurityGroup"/>
        <s:enumeration value="SharePointGroup"/>
        <s:enumeration value="All"/>
      </s:restriction>
    </s:simpleType>
  </s:list></s:simpleType>
```

The following table specifies the possible values for this simple type.

Value	Meaning
None	The principal (1) type is not set.
User	The principal (1) is a user.
DistributionList	The principal (1) is a distribution list .
SecurityGroup	The principal (1) is a security group .
SharePointGroup	The principal (1) is a site (2) group .
All	The entire preceding principal (1) types.

2.2.6 Attributes

This specification does not define any common **XML schema** attribute definitions.

2.2.7 Groups

This specification does not define any common **XML schema** group definitions.

2.2.8 Attribute Groups

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

3 Protocol Details

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

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

Except where specified, protocol clients SHOULD interpret HTTP status codes returned by the protocol server as specified in [\[RFC2616\]](#).

This protocol enables protocol servers to perform implementation-specific authorization checks and to notify protocol clients of authorization faults, either using HTTP status codes or using SOAP faults as specified previously in this section.

3.1 Protocol 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.

The protocol server maintains the following:

- A list of all possible principals (1) that are candidates for resolution. This is used to resolve the user input against a list of possible candidate principals (1). This list MUST NOT be altered by a user of the protocol.
- A list of site collections. This list MUST NOT be altered by a user of the protocol.
- For each site collection, a list of principals (1) that are allowed to use the site collection.

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 WSDL operations as defined by this specification:

Operations	Description
------------	-------------

Operations	Description
ResolvePrincipals	Resolves a list of principals (1) based on a list of criteria.
SearchPrincipals	Finds a list of principals (1) that match the search criteria uniquely or partially.
IsClaimsMode	Determines whether the web application is in claims mode

3.1.4.1 ResolvePrincipals

The **ResolvePrincipals** operation is used to resolve principals (1) based on a list of keys, where each key corresponds to one and only one principal (1) to be resolved, and the principal (1) type as specified in **SPPPrincipalType** (section [2.2.5.1](#)). This operation also provides an option to add the principals (1), if not already present, to the list of principals (1) associated with the site collection.

```
<wsdl:operation name="ResolvePrincipals">
  <wsdl:input message="tns:ResolvePrincipalsSoapIn" />
  <wsdl:output message="tns:ResolvePrincipalsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **ResolvePrincipalsSoapIn** request message, and the protocol server responds with a **ResolvePrincipalsSoapOut** response message, as follows:

1. The caller of this operation **MUST** specify the following criteria: the list of keys (which can be any string), the principal (1) type, and a Boolean indicating whether the user **MUST** be added to the site collection list of principals (1).
2. This operation **MUST** return the principals (1) that match the criteria specified in the request uniquely or partially. Unique match means there is a unique principal whose name, display name, e-mail address, or **Session Initiation Protocol (SIP) address** exactly matches the input (case-insensitive). Partial match means one of those fields starts with the input. The number of uniquely matched principals (1) returned **MUST** be less than or equal to the number of keys provided as input.

This operation **MUST** return a SOAP Fault element in the following cases:

- The *principalKeys* parameter is NULL.
- A *string* element in the *principalKeys* parameter is NULL.

If the user does not have permission to browse for principals (1) this operation **MUST** return an HTTP status code of 403 when using **forms authentication**.[<2>](#)

3.1.4.1.1 Messages

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

Message	Description
ResolvePrincipalsSoapIn	A request to initiate a ResolvePrincipals operation on the protocol server.
ResolvePrincipalsSoapOut	A response from the protocol server at completion of the ResolvePrincipals operation.

3.1.4.1.1.1 ResolvePrincipalsSoapIn

The **ResolvePrincipalsSoapIn** message is the request message for a **ResolvePrincipals** operation.

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

```
http://schemas.microsoft.com/sharepoint/soap/ResolvePrincipals
```

The **SOAP body** contains a **ResolvePrincipals** element. This message MUST contain the following criteria: the *principalKeys*, the principal (1) type, and *addToUserInfoList*, a Boolean value indicating whether the principal (1) MUST be added to the list of principals (1) associated with the site collection. Only a principal's (1) information found in the *AccountName*, *Email*, and *DisplayName* of its **PrincipalInfo** fields (section [2.2.4.2](#)) is included when it is added to the list of principals (1) on the site collection.

3.1.4.1.1.2 ResolvePrincipalsSoapOut

The **ResolvePrincipalsSoapOut** message is the response message for a **ResolvePrincipals** operation (section [3.1.4.1](#)).

The SOAP body contains a **ResolvePrincipalsResponse** element. The value of this element MUST be the list of principals (1) that match the criteria specified in the request uniquely or partially.

3.1.4.1.2 Elements

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

Element	Description
ResolvePrincipals	Body of the ResolvePrincipalsSoapIn message.
ResolvePrincipalsResponse	Body of the ResolvePrincipalsSoapOut message.

3.1.4.1.2.1 ResolvePrincipals

The **ResolvePrincipals** element provides the input data for a **ResolvePrincipals** operation.

```
<s:element name="ResolvePrincipals">
  <s:complexType>
    <s:sequence>
      <s:element name="principalKeys" type="tns:ArrayOfString" minOccurs="0" maxOccurs="1"/>
      <s:element name="principalType" type="tns:SPPrincipalType" minOccurs="1"
maxOccurs="1"/>
      <s:element name="addToUserInfoList" type="s:boolean" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

principalKeys: The list of keys used to match the principals (1). The key SHOULD be the user name or e-mail, or the display name of the principal (1).

principalType: The principal (1) type. This value MUST be one of the **SPPincipalType** values (section [2.2.5.1](#)).

addToUserInfoList: A Boolean that specifies whether the principal (1) MUST be added to the user's site collection if it is uniquely identified and it does not already exist in the user's site collection. If this element is set to **true**, the principal (1) MUST be added to the user's site collection if it is uniquely identified and it does not already exist in the user's site collection. [<3>](#) If this element is set to **false**, the user MUST NOT be added to the site collection list of users.

3.1.4.1.2.2 ResolvePrincipalsResponse

The ResolvePrincipalsResponse element provides the result data for a **ResolvePrincipals** operation.

```
<s:element name="ResolvePrincipalsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ResolvePrincipalsResult" type="tns:ArrayOfPrincipalInfo" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

ResolvePrincipalsResult: The result data for a **ResolvePrincipals** operation. This element's value MUST be the list of **PrincipalInfo** complex types (section [2.2.4.2](#)) that match the criteria specified in the request uniquely or partially. The number of **PrincipalInfo** complex types (section [2.2.4.2](#)) returned MUST be equal to the number of keys provided as input. This element MUST be present.

Each **PrincipalInfo** complex type (section [2.2.4.2](#)) returned MUST have the following fields:

- **AccountName:** If **IsResolved** is set to **true**, the value of this element MUST be the resolved account name. If **IsResolved** is set to **false**, the value MUST be the key provided as input for **ResolvePrincipals** (section [3.1.4.1](#)) operation.
- **UserInfoID:** This value MUST be greater than or equal to zero if the user is already a member of the site collection. If the user cannot be resolved, the value MUST be -1. The value of this parameter MUST NOT be less than -1.
- **IsResolved:** This value MUST be set to **true** if the principal (1) is uniquely identified in the site collection list of principals (1). Otherwise, it MUST be set to **false**. If **IsResolved** is set to **FALSE**, the value of **UserInfoID** MUST be -1.
- **MoreMatches:** If **IsResolved** is set to **true**, this element MUST be NULL. If **IsResolved** is set to **FALSE**, this element contains the list of principals (1) that partially match the input criteria. The list can be empty if there are no partially matching principals (1). This list MUST contain no more than 10 principals (1), even if there are more than 10 principals (1) that partially satisfy the criteria.
- **PrincipalType:** If **IsResolved** is set to **true**, this value MUST NOT be set to All. If **IsResolved** is set to **false**, this value is the **PrincipalType** provided as input for **ResolvePrincipals** operation (section [3.1.4.1](#)).
- **DisplayName:** If **IsResolved** is set to **false**, this element MUST be NULL. If **IsResolved** is set to **true**, this element can be NULL, empty, or contain the resolved principal's (1) **DisplayName** field.
- **Title:** If **IsResolved** is set to **false**, this element MUST be NULL. If **IsResolved** is set to **true**, this element can be NULL, empty, or contain the resolved principal's (1) **Title** field.
- **Department:** If **IsResolved** is set to **false**, this element MUST be NULL. If **IsResolved** is set to **true**, this element can be NULL, empty, or contain the resolved principal's (1) **Department** field.

- **Email:** If **IsResolved** is set to **false**, this element MUST be NULL. If **IsResolved** is set to **true**, this element can be NULL, empty, or contain the resolved principal's (1) **Email** field.

3.1.4.1.3 Complex Types

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

Complex type	Description
ArrayOfString	Array of string elements used to specify the list of principals.

3.1.4.1.3.1 ArrayOfString

This complex type contains an array of string elements. It is used to specify the list of principals (1) in the **ResolvePrincipals** operation.

```
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element name="string" type="s:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

string: A key used to match a principal (1).

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 SearchPrincipals

The **SearchPrincipals** operation finds the principals (1) matching the specified search criteria and the principal (1) type. The protocol client MUST specify the upper limit for the number of uniquely or partially matched principals (1) returned by this operation.

```
<wsdl:operation name="SearchPrincipals">
  <wsdl:input message="tns:SearchPrincipalsSoapIn" />
  <wsdl:output message="tns:SearchPrincipalsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **SearchPrincipalsSoapIn** request message, and the protocol server responds with a **SearchPrincipalsSoapOut** response message, as follows:

1. The caller of this operation MUST specify search criteria, a search text (which can be any string), a principal (1) type, and the upper limit for the number of uniquely or partially matched principals (1).
2. This operation MUST return the principals (1) that match the criteria specified in the request uniquely or partially up to the specified limit. Unique match means there is a unique principal whose name, display name, e-mail address, or **Session Initiation Protocol (SIP) address** exactly matches the input (case-insensitive). Partial match means one of those fields starts with the input.

This operation MUST return a SOAP Fault element if the search text is NULL.

If the user does not have permission to browse for principals (1) this operation MUST return an HTTP status code of 403 if using **forms authentication**.[<4>](#)

3.1.4.2.1 Messages

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

Message	Description
SearchPrincipalsSoapIn	A request to initiate a SearchPrincipals operation on the protocol server.
SearchPrincipalsSoapOut	A response from the protocol server at completion of the SearchPrincipals operation.

3.1.4.2.1.1 SearchPrincipalsSoapIn

The **SearchPrincipalsSoapIn** message is the request message for a **SearchPrincipals** (section [3.1.4.2](#)) operation.

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

```
http://schemas.microsoft.com/sharepoint/soap/SearchPrincipals
```

The SOAP body contains a **SearchPrincipals** element. This message MUST contain the search criteria, a principal (1) type, and the upper limit for the number of uniquely or partially matched principals (1).

3.1.4.2.1.2 SearchPrincipalsSoapOut

The **SearchPrincipalsSoapOut** message is the response message for a **SearchPrincipals** (section [3.1.4.2](#)) operation.

The SOAP body contains a **SearchPrincipalsResponse** (section [3.1.4.2.2.2](#)) element. The value of this element MUST be the list of principals (1) that match the criteria specified in the request uniquely or partially. Unique match means there is a unique principal whose name, display name, e-mail address, or **Session Initiation Protocol (SIP) address** exactly matches the input (case-insensitive). Partial match means one of those fields starts with the input.

3.1.4.2.2 Elements

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

Element	Description
SearchPrincipals	Body of the SearchPrincipalsSoapIn message.
SearchPrincipalsResponse	Body of the SearchPrincipalsSoapOut message.

3.1.4.2.2.1 SearchPrincipals

The **SearchPrincipals** element provides the input data for a **SearchPrincipals** (section [3.1.4.2](#)) operation.

```
<s:element name="SearchPrincipals">
  <s:complexType>
    <s:sequence>
      <s:element name="searchText" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="maxResults" type="s:int" minOccurs="1" maxOccurs="1"/>
      <s:element name="principalType" type="tns:SPPrincipalType" minOccurs="1" maxOccurs="1"
    />
    </s:sequence>
  </s:complexType>
</s:element>
```

searchText: The search string used to find the principals (1). The *searchText* and the *principalType* together comprise the search criteria.

maxResults: The maximum number of principals (1) that can be returned by this operation. If the value of this element is less than zero, an empty list of principals (1) MUST be returned.

PrincipalType: The type of principals (1) that need to be found. This element's value MUST be one of the **SPPrincipalType** (section [2.2.5.1](#)) values.

3.1.4.2.2.2 SearchPrincipalsResponse

The **SearchPrincipalsResponse** element provides the result data for a **SearchPrincipals** operation.

```
<s:element name="SearchPrincipalsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SearchPrincipalsResult" type="tns:ArrayOfPrincipalInfo" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

SearchPrincipalsResult: The list of principals (1), in the format of **PrincipalInfo** (section 2.2.4.2) complex types, which match the specified criteria in the request uniquely or partially. This element MUST be present.

Each **PrincipalInfo** (section 2.2.4.2) returned MUST have the following fields:

UserInfoID: This value MUST NOT be less than -1.

IsResolved: This value MUST be set to **true**.

MoreMatches: This value MUST be NULL.

AccountName: This value MUST NOT be NULL.

PrincipalType: This value MUST NOT be set to All.

DisplayName: This element can be NULL, empty, or contain the resolved principal's (1) display name field.

Title: This element can be NULL, empty, or contain the resolved principal's (1) title field.

Department: This element can be NULL, empty, or contain the resolved principal's (1) department field.

Email: This element can be NULL, empty, or contain the resolved principal's (1) **email** field.

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 IsClaimsMode

The **IsClaimsMode** operation is used to determine whether the claims service is used to search and resolve the principals (1).

```
<wsdl:operation name="IsClaimsMode">
  <wsdl:input message="tns:IsClaimsModeSoapIn" />
  <wsdl:output message="tns:IsClaimsModeSoapOut" />
</wsdl:operation>
```

The protocol client sends an **IsClaimsModeSoapIn** request message, and the protocol server responds with an **IsClaimsModeSoapOut** response message, as follows:

- the operation MUST return a Boolean value representing whether the web application is in claims mode.

3.1.4.3.1 Messages

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

Message	Description
IsClaimsModeSoapIn	A request to initiate an IsClaimsMode operation on the protocol server.
IsClaimsModeSoapOut	A response from the protocol server at completion of the IsClaimsMode operation.

3.1.4.3.1.1 IsClaimsModeSoapIn

The **IsClaimsModeSoapIn** message is the request message for an **IsClaimsMode** operation.

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

```
http://schemas.microsoft.com/sharepoint/soap/IsClaimsMode
```

The SOAP body contains an **IsClaimsMode** element.

3.1.4.3.1.2 IsClaimsModeSoapOut

The **IsClaimsModeSoapOut** message is the response message for an **IsClaimsMode** operation.

The SOAP body contains an **IsClaimsModeResponse** element. The value of this element **MUST** contain the claims mode result.

3.1.4.3.2 Elements

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

Element	Description
IsClaimsMode	Body of the IsClaimsModeSoapIn message.
IsClaimsModeResponse	Body of the IsClaimsModeSoapOut message.

3.1.4.3.2.1 IsClaimsMode

The **IsClaimsMode** element provides the input data for an **IsClaimsMode** (section [3.1.4.2](#)) operation.

```
<s:element name="IsClaimsMode">
  <s:complexType />
</s:element>
```

3.1.4.3.2.2 IsClaimsModeResponse

The **IsClaimsModeResponse** element provides the result data for an **IsClaimsMode** operation.

```
<s:element name="IsClaimsModeResponse">
  <s:complexType>
    <s:sequence>
```

```
        <s:element minOccurs="1" maxOccurs="1" name="IsClaimsModeResult" type="s:boolean"
/>
    </s:sequence>
</s:complexType>
</s:element>
```

IsClaimsModeResult: This value MUST NOT be NULL

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following subsections contain examples of usage of this protocol.

4.1 Resolve Principals Knowing Their E-mail Addresses

In this scenario, the protocol client resolves a list of principals (1) knowing their e-mail addresses. The protocol client is interested in all types of principals (1) and it does not attempt to add them to the site collection list of users.

The protocol client issues a **ResolvePrincipals** request with the known e-mail addresses as principal keys, the principal (1) type set to ALL and the *addToUserInfoList* parameter set to **false**. The protocol client sends the following WSDL message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ResolvePrincipals xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <principalKeys>
        <string>ben@contoso.com</string>
        <string>doesnotexist@contoso.com</string>
      </principalKeys>
      <principalType>All</principalType>
      <addToUserInfoList>false</addToUserInfoList>
    </ResolvePrincipals>
  </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/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ResolvePrincipalsResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <ResolvePrincipalsResult>
        <PrincipalInfo>
          <AccountName>MYDOMAIN\account1</AccountName>
          <UserInfoID>1</UserInfoID>
          <DisplayName>Ben Smith</DisplayName>
          <Email>ben@contoso.com</Email>
          <Department>Marketing</Department>
          <Title/>
          <IsResolved>true</IsResolved>
          <PrincipalType>User</PrincipalType>
        </PrincipalInfo>
        <PrincipalInfo>
          <AccountName>doesnotexist@contoso.com</AccountName>
          <UserInfoID>-1</UserInfoID>
          <IsResolved>false</IsResolved>
          <MoreMatches />
          <PrincipalType>All</PrincipalType>
        </PrincipalInfo>
      </ResolvePrincipalsResult>
```



```

    </ResolvePrincipalsResponse>
  </soap:Body>
</soap:Envelope>

```

The protocol client searches the returned array with information about the principals (1) and checks whether they were resolved.

4.2 Search for Principals Knowing the E-mail Address

In this scenario, the protocol client searches for a maximum of 15 principals (1) that have an e-mail address containing "marketing." The protocol client is interested in all types of principals (1).

The protocol client issues a **SearchPrincipals** request with the e-mail prefix of "marketing" as the searchText, the principal (1) type set to ALL, and 15 as the maximum value of principals (1) to return. The protocol client sends the following WSDL message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SearchPrincipals xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <searchText>marketing</searchText>
      <maxResults>15</maxResults>
      <principalType>All</principalType>
    </SearchPrincipals>
  </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/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SearchPrincipalsResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <SearchPrincipalsResult>
        <PrincipalInfo>
          <AccountName>MYDOMAIN\account2</AccountName>
          <UserInfoID>-1</UserInfoID>
          <DisplayName>Marketing - West</DisplayName>
          <Email>marketing-west@contoso.com</Email>
          <Department>Marketing</Department>
          <Title />
          <IsResolved>>true</IsResolved>
          <PrincipalType>User</PrincipalType>
        </PrincipalInfo>
        <PrincipalInfo>
          <AccountName>MYDOMAIN\account3</AccountName>
          <UserInfoID>-1</UserInfoID>
          <DisplayName>Marketing Communication List</DisplayName>
          <Email>mcl@contoso.com</Email>
          <Department />
          <IsResolved>>true</IsResolved>
          <PrincipalType>DistributionList</PrincipalType>

```

```

        </PrincipalInfo>
    </SearchPrincipalsResult>
</SearchPrincipalsResponse>
</soap:Body>
</soap:Envelope>

```

The returned array contains no more than 15 principals (1) that match "marketing." Because the client in this scenario is searching exclusively for principals (1) that have an e-mail address containing "marketing," the client will perform additional processing on the array to further limit the results.

4.3 Check Whether in Claims Mode

In this scenario, the protocol client determines whether the claims service is used for the **ResolvePrincipals** and **SearchPrincipals** operations. The protocol client sends the following WSDL message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <IsClaimsMode xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      </IsClaimsMode>
    </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/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <IsClaimsModeResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <IsClaimsModeResult>false</IsClaimsModeResult>
    </IsClaimsModeResponse>
  </soap:Body>
</soap:Envelope>

```

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 and schema are provided in this appendix.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/">
      <s:element name="IsClaimsMode">
        <s:complexType />
      </s:element>
      <s:element name="IsClaimsModeResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="IsClaimsModeResult" type="s:boolean"
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ResolvePrincipals">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="principalKeys"
type="tns:ArrayOfString" />
            <s:element minOccurs="1" maxOccurs="1" name="principalType"
type="tns:SPPrincipalType" />
            <s:element minOccurs="1" maxOccurs="1" name="addToUserInfoList" type="s:boolean"
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfString">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:simpleType name="SPPrincipalType">
        <s:list>
          <s:simpleType>
            <s:restriction base="s:string">
              <s:enumeration value="None" />
              <s:enumeration value="User" />
              <s:enumeration value="DistributionList" />
              <s:enumeration value="SecurityGroup" />
              <s:enumeration value="SharePointGroup" />
              <s:enumeration value="All" />
            </s:restriction>
          </s:simpleType>
        </s:list>
      </s:simpleType>
      <s:element name="ResolvePrincipalsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ResolvePrincipalsResult"
type="tns:ArrayOfPrincipalInfo" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>

```

```

        </s:complexType>
    </s:element>
    <s:complexType name="ArrayOfPrincipalInfo">
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="unbounded" name="PrincipalInfo"
type="tns:PrincipalInfo" />
        </s:sequence>
    </s:complexType>
    <s:complexType name="PrincipalInfo">
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="AccountName" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="UserInfoID" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Email" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Department" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Title" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="IsResolved" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="MoreMatches"
type="tns:ArrayOfPrincipalInfo" />
            <s:element minOccurs="1" maxOccurs="1" name="PrincipalType"
type="tns:SPPrincipalType" />
        </s:sequence>
    </s:complexType>
    <s:element name="SearchPrincipals">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="searchText" type="s:string" />
                <s:element minOccurs="1" maxOccurs="1" name="maxResults" type="s:int" />
                <s:element minOccurs="1" maxOccurs="1" name="principalType"
type="tns:SPPrincipalType" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="SearchPrincipalsResponse">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="SearchPrincipalsResult"
type="tns:ArrayOfPrincipalInfo" />
            </s:sequence>
        </s:complexType>
    </s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="IsClaimsModeSoapIn">
    <wsdl:part name="parameters" element="tns:IsClaimsMode" />
</wsdl:message>
<wsdl:message name="IsClaimsModeSoapOut">
    <wsdl:part name="parameters" element="tns:IsClaimsModeResponse" />
</wsdl:message>
<wsdl:message name="ResolvePrincipalsSoapIn">
    <wsdl:part name="parameters" element="tns:ResolvePrincipals" />
</wsdl:message>
<wsdl:message name="ResolvePrincipalsSoapOut">
    <wsdl:part name="parameters" element="tns:ResolvePrincipalsResponse" />
</wsdl:message>
<wsdl:message name="SearchPrincipalsSoapIn">
    <wsdl:part name="parameters" element="tns:SearchPrincipals" />
</wsdl:message>
<wsdl:message name="SearchPrincipalsSoapOut">
    <wsdl:part name="parameters" element="tns:SearchPrincipalsResponse" />
</wsdl:message>
<wsdl:portType name="PeopleSoap">
    <wsdl:operation name="IsClaimsMode">
        <wsdl:input message="tns:IsClaimsModeSoapIn" />
        <wsdl:output message="tns:IsClaimsModeSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="ResolvePrincipals">
        <wsdl:input message="tns:ResolvePrincipalsSoapIn" />
        <wsdl:output message="tns:ResolvePrincipalsSoapOut" />
    </wsdl:operation>

```

```

    </wsdl:operation>
    <wsdl:operation name="SearchPrincipals">
      <wsdl:input message="tns:SearchPrincipalsSoapIn" />
      <wsdl:output message="tns:SearchPrincipalsSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="PeopleSoap" type="tns:PeopleSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="IsClaimsMode">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/IsClaimsMode"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ResolvePrincipals">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ResolvePrincipals" style="document"
/>
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SearchPrincipals">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/SearchPrincipals" 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="PeopleSoap12" type="tns:PeopleSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="IsClaimsMode">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/IsClaimsMode" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ResolvePrincipals">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ResolvePrincipals" style="document"
/>
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SearchPrincipals">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/SearchPrincipals" style="document"
/>
      <wsdl:input>

```

```
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

7 Appendix B: Product Behavior

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

- The 2007 Microsoft Office system
- Microsoft Office 2010 suites
- Microsoft Office 2013
- Windows SharePoint Services 3.0
- Microsoft SharePoint Foundation 2010
- Microsoft SharePoint Foundation 2013
- Microsoft Windows
- Microsoft Office 2016
- Microsoft SharePoint Server 2016
- Microsoft Office 2019
- Microsoft SharePoint Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

[<1> Section 2.2.4.2](#): The login name is in the format: Domain\UserName for Windows accounts. It is in the format Provider:UserName for a SharePoint membership account, where **Provider** is the name of the membership provider.

[<2> Section 3.1.4.1](#): When using Windows authentication, if the user does not have permission to browse for principals (1) then this operation MUST return an HTTP status code of 401.

[<3> Section 3.1.4.1.2.1](#): SharePoint Foundation 2010 and SharePoint Foundation 2013 do not add the principal to the site collection if it is of the type Distribution List.

[<4> Section 3.1.4.2](#): If the user does not have permission to browse for principals (1) this operation MUST return an HTTP status code of 401 if using Windows authentication.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

Section	Description	Revision class
Z Appendix B: Product Behavior	Updated list of supported products.	Major

9 Index

A

Abstract data model
 [server](#) 14
[Applicability](#) 8
[ArrayOfPrincipalInfo complex type](#) 11
[Attribute groups](#) 13
[Attributes](#) 13

C

[Capability negotiation](#) 8
[Change tracking](#) 33
[Check whether in claims mode example](#) 26
[Complex types](#) 11
 [ArrayOfPrincipalInfo](#) 11
 [PrincipalInfo](#) 11

D

Data model - abstract
 [server](#) 14

E

Events
 [local - server](#) 23
 [timer - server](#) 23
Examples
 [check whether in claims mode](#) 26
 [overview](#) 24
 [resolve principals knowing their e-mail addresses](#)
 24
 [search for principals knowing the e-mail address](#)
 25

F

[Fields - vendor-extensible](#) 8
[Full WSDL](#) 28

G

[Glossary](#) 6
[Groups](#) 13

I

[Implementer - security considerations](#) 27
[Index of security parameters](#) 27
[Informative references](#) 7
Initialization
 [server](#) 14
[Introduction](#) 6
[IsClaimsMode operation](#) 21
 [attribute groups](#) 23
 [attributes](#) 23
 [complex types](#) 23
 [elements](#) 22
 [IsClaimsMode](#) 22
 [IsClaimsModeResponse](#) 22

[groups](#) 23
 [messages](#) 21
 [IsClaimsModeSoapIn](#) 22
 [IsClaimsModeSoapOut](#) 22
 [simple types](#) 23

L

Local events
 [server](#) 23

M

Message processing
 [server](#) 14
Messages
 [ArrayOfPrincipalInfo complex type](#) 11
 [attribute groups](#) 13
 [attributes](#) 13
 [complex types](#) 11
 [elements](#) 11
 [enumerated](#) 10
 [groups](#) 13
 [namespaces](#) 10
 [PrincipalInfo complex type](#) 11
 [simple types](#) 12
 [SPPrincipalType simple type](#) 12
 [syntax](#) 10
 [transport](#) 10

N

[Namespaces](#) 10
[Normative references](#) 7

O

Operations
 [IsClaimsMode](#) 21
 [ResolvePrincipals](#) 15
 [SearchPrincipals](#) 18
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 27
[Preconditions](#) 8
[Prerequisites](#) 8
[PrincipalInfo complex type](#) 11
[Product behavior](#) 32
Protocol Details
 [overview](#) 14

R

[References](#) 7
 [informative](#) 7
 [normative](#) 7
[Relationship to other protocols](#) 8
[Resolve principals knowing their e-mail addresses](#)
 [example](#) 24

- [ResolvePrincipals operation](#) 15
 - [attribute groups](#) 18
 - [attributes](#) 18
 - [complex types](#) 18
 - [ArrayOfString](#) 18
 - [elements](#) 16
 - [ResolvePrincipals](#) 16
 - [ResolvePrincipalsResponse](#) 17
 - [groups](#) 18
 - [messages](#) 15
 - [ResolvePrincipalsSoapIn](#) 16
 - [ResolvePrincipalsSoapOut](#) 16
 - [simple types](#) 18

S

- [Search for principals knowing the e-mail address](#)

- [example](#) 25

- [SearchPrincipals operation](#) 18

- [attribute groups](#) 21
 - [attributes](#) 21
 - [complex types](#) 21
 - [elements](#) 19
 - [SearchPrincipals](#) 20
 - [SearchPrincipalsResponse](#) 20
 - [groups](#) 21
 - [messages](#) 19
 - [SearchPrincipalsSoapIn](#) 19
 - [SearchPrincipalsSoapOut](#) 19
 - [simple types](#) 21

- Security

- [implementer considerations](#) 27
 - [parameter index](#) 27

- Sequencing rules

- [server](#) 14

- Server

- [abstract data model](#) 14
 - [initialization](#) 14
 - [IsClaimsMode operation](#) 21
 - [local events](#) 23
 - [message processing](#) 14
 - [ResolvePrincipals operation](#) 15
 - [SearchPrincipals operation](#) 18
 - [sequencing rules](#) 14
 - [timer events](#) 23
 - [timers](#) 14
- [Simple types](#) 12
 - [SPPrincipalType](#) 12
- [SPPrincipalType simple type](#) 12
- [Standards assignments](#) 9

- Syntax

- [messages - overview](#) 10

T

- Timer events

- [server](#) 23

- Timers

- [server](#) 14

- [Tracking changes](#) 33

- [Transport](#) 10

- Types

- [complex](#) 11
 - [simple](#) 12

V

- [Vendor-extensible fields](#) 8

- [Versioning](#) 8

W

- [WSDL](#) 28