[MS-DLX]:

Distribution List Expansion Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial version
4/25/2008	0.2		Revised and edited the technical feedback
6/27/2008	1.0		Revised and edited the technical feedback
8/15/2008	1.01		Revised and edited the technical feedback
12/12/2008	2.0		Revised and edited the technical feedback
2/13/2009	2.01		Revised and edited the technical feedback
3/13/2009	2.02		Edited the technical feedback
7/13/2009	2.03	Major	Revised and edited the technical content
8/28/2009	2.04	Editorial	Revised and edited the technical content
11/6/2009	2.05	Editorial	Revised and edited the technical content
2/19/2010	2.06	Editorial	Revised and edited the technical content
3/31/2010	2.07	Major	Updated and revised the technical content
4/30/2010	2.08	Editorial	Revised and edited the technical content
6/7/2010	2.09	Editorial	Revised and edited the technical content
6/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.10	No Change	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	4.0	Major	Significantly changed the technical content.
4/11/2012	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.0.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	4.0.1	No Change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/30/2013	4.1	Minor	Clarified the meaning of the technical content.
11/18/2013	4.1	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	4.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	4.2	Minor	Clarified the meaning of the technical content.
7/31/2014	4.2	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	4.3	Minor	Clarified the meaning of the technical content.
3/30/2015	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the procedure for expanding **distribution lists**. It specifies the **web service method** that is used to get the **membership** of a distribution list. The same Web service can also be used to search for users and distribution lists and query attributes associated with each. The web service can also serve as a proxy for searching users in the **Skype public directory**.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

address book: A collection of Address Book objects, each of which are contained in any number of address lists.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also Active Directory.

display name: A text string that is used to identify a principal or other object in the user interface. Also referred to as title.

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

fully qualified domain name (FQDN): An unambiguous domain name (2) that gives an absolute location in the Domain Name System's (DNS) hierarchy tree, as defined in [RFC1035] section 3.1 and [RFC2181] section 11.

Global Address List (GAL): An address list that conceptually represents the default address list for an **address book**.

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.

Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS): An extension of **HTTP** that securely encrypts and decrypts webpage requests.

Kerberos: An authentication (2) system that enables two parties to exchange private information across an otherwise open network by assigning a unique key (called a ticket) to each user that logs on to the network and then embedding these tickets into messages sent by the users. For more information, see [MS-KILE].

membership: The state or status of being a member of a member group. A membership contains additional metadata such as the privacy level that is associated with the membership.

NT LAN Manager (NTLM) Authentication Protocol: A protocol using a challenge-response mechanism for authentication (2) in which clients are able to verify their identities without sending a password to the server. It consists of three messages, commonly referred to as Type 1 (negotiation), Type 2 (challenge) and Type 3 (authentication). For more information, see [MS-NLMP].

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data—a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client authentication (2) using X.509 certificates (2). For more information, see [X509]. The SSL protocol is precursor to **Transport Layer Security (TLS)**. The TLS version 1.0 specification is based on SSL version 3.0.

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

Skype public directory: A directory containing a list of all Skype consumer users.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

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 envelope: A container for **SOAP message** information and the root element of a **SOAP** document. See [SOAP1.2-1/2007] section 5.1 for more information.

SOAP message: An XML document consisting of a mandatory **SOAP envelope**, an optional SOAP header, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). **TLS** is standardized in the IETF TLS working group. See [RFC4346].

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

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

web service method: A procedure that is exposed to web service clients as an operation that can be called on the web service. Also referred to as web method.

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

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

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

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

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-ABS] Microsoft Corporation, "Address Book File Structure".

[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

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

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, June 2002, http://www.ietf.org/rfc/rfc3261.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/

1.2.2 Informative References

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication Protocol".

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, http://www.rfc-editor.org/rfc/rfc4559.txt

1.3 Overview

This protocol is used to expand distribution lists or to search for users and distribution lists and query attributes associated with each one. Using this protocol, a user can provide the address of a distribution list and obtain information about its membership. The user can also provide a search string, a set of attributes to search against, and a set of attributes to return for each match. The search can return one or more users or distribution lists or both and get attributes about each one found. The protocol can also be used for searching users in the Skype public directory, and provide feedback on the corresponding search results.

The protocol consists of one request and one response. The request contains either a distribution list expansion request, a search request, or search feedback data. The request contains the information needed to describe the request. The response contains the response status and, if the response is successful, the data requested.

This protocol is conceptually three Web services methods. This documentation specifies the structure of the schema used to construct the body in the request and response messages. The protocol uses **Simple Object Access Protocol (SOAP)** and **Web Services Description Language (WSDL)** to describe the structure of the message body. The full WSDL is included in section 6.

1.4 Relationship to Other Protocols

This protocol uses SOAP over **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818], as shown in the following layering diagram:

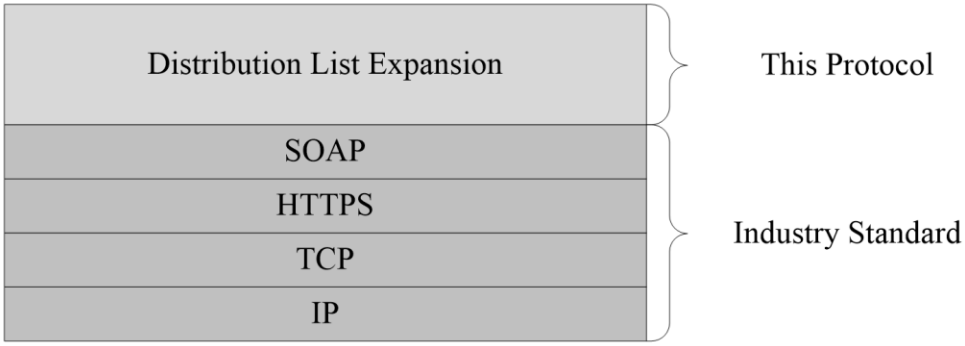

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

For a client that uses this protocol with a server, it is assumed that the server has an operational SOAP1.2/HTTP1.1/TCP/IP stack, as described in [RFC2616]. It is also assumed that the client has the **fully qualified domain name (FQDN)** of the server to which the client will connect. The client can obtain the FQDN of the server via a different channel, for example, in the **Session Initiation Protocol (SIP)** signaling channel, as described in [RFC3261]. The server also requires that the client be able to negotiate **Hypertext Transfer Protocol (HTTP)** over **Transport Layer Security (TLS)** to establish the connection.

1.6 Applicability Statement

This protocol is applicable for

- Expanding distribution lists
- Searching for users and distribution lists and querying attributes associated with each one.
- Searching for users in the Skype public directory, and provide feedback on the search results.

1.7 Versioning and Capability Negotiation

None.

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

This protocol MUST use HTTPS, as specified in [RFC2818], over **Transmission Control Protocol (TCP)** as transport. The HTTP traffic MUST be encrypted with **Secure Sockets Layer (SSL)**. The client can obtain the address to connect to the server via a different channel, such as SIP.

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] and [XMLSCHEMA2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

This protocol 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
soap	http://schemas.xmlsoap.org/wsdl/soap	SOAP [SOAP1.1]
s	http://www.w3.org/2001/XMLSchema	XML Schema [XMLSCHEMA1], [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12	SOAP1.2 [SOAP1.2/1], [SOAP1.2/2]
http	http://schemas.xmlsoap.org/wsdl/http	HTTP [RFC2616]
wsdl	http://schemas.xmlsoap.org/wsdl	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

This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types

This specification does not define any common XML schema simple type definitions.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

2.2.9 Common Data Structures

This specification does not define any common XML schema data structures.

3 Protocol Details

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

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

3.1 Server Details

3.1.1 Abstract Data Model

This 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.

A **directory service (DS)**<1> is assumed on the server side. Users and distribution lists are usually stored as objects in the directory service. The protocol enables a client to query attributes about both types of objects.

In the sections that follow, reference is made to attribute names. The attribute names in the following table MUST be implemented by the server.

Attribute Name	Description
EntryId	Unique string that identifies a particular user or distribution list. This value MUST not change even when a user or distribution list is renamed or moved. It changes if a user or distribution list is deleted and recreated.
AbEntryHash	Unique string that identifies the current state of an entry. If one of its attributes changes value or an attribute is added or deleted, the user or distribution list's **AbEntryHash** value MUST change.
PhotoRelPath	Relative path to the photo file. The path MUST be relative to the file system folder described in [MS-ABS] section 1.3.
PhotoSize	Size of the photo file in bytes.
PhotoHash	Unique string that identifies the current state of an entry's photo. If the photo changes, the **PhotoHash** value MUST change.
OrgHash	Unique string that identifies the current state of an entry's organizational structure. If the structure changes, the **OrgHash** value MUST change.

Additional attribute(s) may be requested, and the name of the attribute will be the same name of the corresponding attribute stored in Active Directory.

3.1.2 Timers

The only timers of concern are the timers for HTTP.

3.1.3 Initialization

As part of initialization, the server MUST start listening for incoming requests on an HTTP **Uniform Resource Locator (URL)**. The client MUST have access to this HTTP URL and can obtain the URL by a channel that is separate from the HTTP channel used for expanding distribution lists, for example, through SIP.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
ExpandDistributionList	This operation expands a distribution list.
ProvideSkypeSearchFeedback	This operation provides feedback on the Skype public directory search results.<2>
SearchAbEntry	This operation searches for a user or distribution list.
SearchSkypeDirectory	This operation searches for a user in the

Operation	Description
	Skype public directory <3>.

3.1.4.1 ExpandDistributionList

The following excerpt from this protocol's WSDL specifies the messages that constitute this operation.

```
<wsdl:operation name="ExpandDistributionList">
  <wsdl:input message="tns:ExpandDistributionListSoapIn" />
  <wsdl:output message="tns:ExpandDistributionListSoapOut" />
</wsdl:operation>
```

When user action triggers a request to expand a distribution list, a TCP connection MUST be made to the server and SSL MUST be negotiated. The address of the server that makes the TCP connection MUST be obtained through a different channel, such as SIP. After successful SSL negotiation, a SOAP HTTP request, **ExpandDistributionListSoapIn** message, MUST be constructed with a **SOAP body** containing the **ExpandDistributionList** element.

On receiving an **ExpandDistributionList** request, the server SHOULD query the repository of distribution lists to get all of the members of the distribution list specified in the request. After obtaining the membership, the server MUST construct the **ExpandDistributionListSoapOut** message, containing the **ExpandDistributionListResponse** element, and it MUST send the message in the SOAP HTTP response, which is a 2xx response to a SOAP HTTP request, as described in [RFC2616]. In the case of errors, the response message MUST specify the error and the type of error that was encountered by the server.

3.1.4.1.1 Messages

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

Message	Description
ExpandDistributionListSoapIn	Request from client to expand a distribution list.
ExpandDistributionListSoapOut	Response from server after it executes a request to expand a distribution list.

3.1.4.1.1.1 ExpandDistributionListSoapIn

The **ExpandDistributionListSoapIn SOAP message** is a request that is sent from the client, typically as a result of a user action to expand a distribution list. The request information MUST be captured in the **ExpandDistributionList** element in the SOAP body of the message. The **ExpandDistributionList** element is specified in section 3.1.4.1.3.1.

3.1.4.1.1.2 ExpandDistributionListSoapOut

The **ExpandDistributionListSoapOut** SOAP message is a response that is sent by the server after it executes a request to expand a distribution list. This message contains the result of the expansion on the server. The result is represented in the **ExpandDistributionListResponse** element, which MUST be in the SOAP body of the SOAP message. The **ExpandDistributionListResponse** element is specified in section 3.1.4.1.3.2. If the server is able to successfully acquire the membership of the distribution list, the response element contains the membership details. These details for each member are included in the complex type **ActiveDirectoryObjectInfo**, which is specified in section 3.1.4.1.3.5. In case of an error, the response element MUST specify the reason for the failure to expand the distribution list in the simple type **ResponseState**, which is specified in section 3.1.4.1.4.1.

3.1.4.1.2 Elements

All elements in the WSDL are contained in complex types and specified in section 3.1.4.1.3.

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
ExpandDistributionList	The overall container of the request to expand a distribution list.
ExpandDistributionListResponse	The overall container of the response to the request to expand a distribution list. **ExpandDistributionListResponse** contains the response status and, if the response is successful, the expanded distribution list that was requested.
DlxGroup	The container for information about the membership of the distribution list.
ArrayOfActiveDirectoryObjectInfo	A sequence of elements of type **ActiveDirectoryObjectInfo**.
ActiveDirectoryObjectInfo	The container for information about an individual member of the distribution list.

3.1.4.1.3.1 ExpandDistributionList

The **ExpandDistributionList** complex type is the overall container of the information that is sent in the SOAP request to expand a distribution list. The schema of the request body within the **SOAP envelope** MUST be as follows.

```
<s:schema elementFormDefault="qualified" targetNamespace="DistributionListExpander">
  <s:element name="ExpandDistributionList">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="groupMailAddress" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

ExpandDistributionList.groupMailAddress: This element is of type string and indicates the name of the distribution list that is to be expanded. There MUST be exactly one such element present in each **ExpandDistributionList** request.

3.1.4.1.3.2 ExpandDistributionListResponse

The **ExpandDistributionListResponse** complex type is the overall container in the response to the **ExpandDistributionList** request. **ExpandDistributionListResponse** encapsulates the results of the operation to expand a distribution list. It contains one **ExpandDistributionListResult** element of type **DlxGroup**. The schema for this complex type within the SOAP envelope is as follows:

```
<s:element name="ExpandDistributionListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="ExpandDistributionListResult" 
type="tns:DlxGroup" />
    </s:sequence>
  </s:complexType>
</s:element>
```

The following element MUST be contained in the **ExpandDistributionListResponse** complex type:

ExpandDistributionListResponse.ExpandDistributionListResult: This element is of type **DlxGroup** and contains information about the results of the operation to expand a distribution list.

3.1.4.1.3.3 DlxGroup

The **DlxGroup** complex type is the container for information about the membership of the distribution list. It contains a sequence of elements where each element MUST be one of **ResponseStatus**, **Users** or **NestedGroups**. The schema for this complex type is as follows:

```
<s:complexType name="DlxGroup">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="ResponseStatus" type="tns:ResponseState" />
    <s:element minOccurs="0" maxOccurs="1" name="Users" 
type="tns:ArrayOfActiveDirectoryObjectInfo" />
    <s:element minOccurs="0" maxOccurs="1" name="NestedGroups" 
type="tns:ArrayOfActiveDirectoryObjectInfo" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

DlxGroup.ResponseStatus: **ResponseStatus** indicates the result of the **ExpandDistributionList** request and MUST be present. There MUST be exactly one **ResponseStatus** element contained in the **ExpandDistributionListResult** element. The **ResponseStatus** element is of the simple type **ResponseState**. The **ResponseState** type is described in section 3.1.4.1.4.1.

DlxGroup.Users: A **Users** element contains information about each of the members of the distribution list. The number of **Users** elements contained in an **ExpandDistributionListResult** element MUST NOT exceed one. The type of the **Users** element is the complex type **ArrayOfActiveDirectoryObjectInfo**.

DlxGroup.NestedGroups: A **NestedGroups** element contains information about any distribution lists that are members of the distribution list that was expanded. The number of **NestedGroups** elements MUST NOT exceed one. The type of the **NestedGroups** element is the complex type **ArrayOfActiveDirectoryObjectInfo**.

3.1.4.1.3.4 ArrayOfActiveDirectoryObjectInfo

The **ArrayOfActiveDirectoryObjectInfo** complex type consists of a sequence of elements of type **ActiveDirectoryObjectInfo**. This type contains directory information about each member of the distribution list to be expanded in the request. The schema for this complex type is as follows:

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

The **ArrayOfActiveDirectoryObjectInfo** element can contain only the following possible element:

ArrayOfActiveDirectoryObjectInfo.ActiveDirectoryObjectInfo: The **ActiveDirectoryObjectInfo** element is a complex type that contains information about a member of a distribution list.

3.1.4.1.3.5 ActiveDirectoryObjectInfo

The **ActiveDirectoryObjectInfo** element is a complex type that contains information about a user. Each element contained in the **Users** and **NestedGroups** elements is of this type. The schema for this complex type is as follows:

```
<s:complexType name="ActiveDirectoryObjectInfo">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="displayName" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="mail" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="mailNickname" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="sipUri" type="s:string" />
  </s:sequence>
</s:complexType>
```

The **ActiveDirectoryObjectInfo** element allows for the following possible elements to be contained in it:

ActiveDirectoryObjectInfo.displayName: A string type that specifies the **display name** of the user or nested group.

ActiveDirectoryObjectInfo.mail: A string type that specifies the mail address of the user or nested group.

ActiveDirectoryObjectInfo.mailNickname: A string type that specifies any nickname that the user or nested group has.

ActiveDirectoryObjectInfo.sipUri: A string type that specifies the SIP **Uniform Resource Identifier (URI)**, as specified in [RFC3261], of the user or nested group.

3.1.4.1.4 Simple Types

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

Simple type	Description
ResponseState	Completion status of the attempted expansion of the distribution list.

3.1.4.1.4.1 ResponseState

ResponseState is a simple type that is an enumeration. The schema for this type is as follows:

```
<s:simpleType name="ResponseState">
  <s:restriction base="s:string">
    <s:enumeration value="Invalid" />
    <s:enumeration value="Success" />
    <s:enumeration value="MemberCountLimitExceeded" />
    <s:enumeration value="NotAuthorized" />
    <s:enumeration value="NotFound" />
    <s:enumeration value="SimultaneousRequestLimitExceeded" />
  </s:restriction>
</s:simpleType>
```

The following table lists the meaning of each enumerated value.

Value	Meaning
Invalid	The server SHOULD return this string in the **ResponseStatus** element when it determines that the address of the distribution list specified by the **groupMailAddress** element in the request is invalid. The server can return this error in the case of other failures not described by the following statuses.
Success	The server MUST return this string in the **ResponseStatus** element when it successfully expands the distribution list specified by the **groupMailAddress** element in the request.
MemberCountLimitExceeded	The server MUST return this string in the **ResponseStatus** element when the count of members in the distribution list specified by the **groupMailAddress** element exceeds a predefined limit on the server<4>.
NotAuthorized	The server MUST return this string in the **ResponseStatus** element when the server determines that the client is not authorized to view the membership of the distribution list specified by the **groupMailAddress** element in the request<5>.
NotFound	The server MUST return this string in the **ResponseStatus** element when the server is not able to find the distribution list specified by the **groupMailAddress** element in the request.
SimultaneousRequestLimitExceeded	The server SHOULD<6> return this string in the **ResponseStatus** element when the limit for the number of simultaneous requests being processed by the server for a particular user is exceeded.

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 SearchAbEntry

The following excerpt from this protocol's WSDL specifies the messages that constitute this operation:

```
<wsdl:operation name="SearchAbEntry">
  <wsdl:input message="tns:SearchAbEntrySoapIn" />
  <wsdl:output message="tns:SearchAbEntrySoapOut" />
</wsdl:operation>
```

When user action triggers a request to search the **address book**, a TCP connection MUST be made to the server and SSL MUST be negotiated. The address of the server that makes the TCP connection MUST be obtained through a separate channel, such as SIP. After successful SSL negotiation, a SOAP HTTP request containing a **SearchAbEntrySoapIn** message MUST be constructed with a SOAP body containing the **SearchAbEntry** element.

On receiving a **SearchAbEntry** request, the server SHOULD query the repository of users and distribution lists to search for the information specified in the request<7>. After conducting the requested search, the server MUST construct the **SearchAbEntrySoapOut** message, containing the **SearchAbEntryResponse** element, and it MUST send the message in the SOAP HTTP response, which is a 2xx response to a SOAP HTTP request. In case of errors, the response message MUST specify the error and the type of error that was encountered by the server.

3.1.4.2.1 Messages

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

Message	Description
SearchAbEntrySoapIn	Request from the client to search the address book.
SearchAbEntrySoapOut	Response from the server after it executes a request to search the address book.

3.1.4.2.1.1 SearchAbEntrySoapIn

The **SearchAbEntrySoapIn** SOAP message is a request that is sent from the client. For example, a **SearchAbEntrySoapIn** message is created as a result of a user action to search for a user or distribution list. The request information MUST be captured in the **SearchAbEntry** element in the SOAP body of the message. The **SearchAbEntry** element is specified in section 3.1.4.2.3.1.

3.1.4.2.1.2 SearchAbEntrySoapOut

This SOAP message is a response that is sent by the server after it executes a request to search the address book. This message contains the result of the search on the server. The result is represented in the **SearchAbEntryResponse** element, which MUST be in the SOAP body of the SOAP message. The **SearchAbEntryResponse** element is specified in section 3.1.4.2.3.8.

3.1.4.2.2 Elements

All elements in the WSDL are contained in complex types and specified in section 3.1.4.2.3.

3.1.4.2.3 Complex Types

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

Complex type	Description
SearchAbEntry	The overall container of the request to search the address book.
AbEntryRequest	Search request container that MUST contain one of **AbEntryRequest.BasicSearchQuery**, **ArrayOfAbEntryRequest.ChangeSearchQuery** or **AbEntryRequest.OrgSearchQuery** and one of **AbEntryRequest.SearchMetadata.**
AbEntryRequest.BasicSearchQuery	Basic search request.
ArrayOfAbEntryRequest.ChangeSearchQuery	A sequence of elements of type **AbEntryRequest.ChangeSearchQuery.**
AbEntryRequest.ChangeSearchQuery	Container for a single change search request.
AbEntryRequest.SearchMetadata	Search metadata that applies to the search.
AbEntryRequest.OrgSearchQuery	Organizational search request.
SearchAbEntryResponse	The overall container of the response to the search request. **SearchAbEntryResponse** contains the response status and, if the response is successful, the users and distribution lists that match the request.
AbEntryResponse	Container for the results of the search.
ArrayOfAbEntry	Sequence of elements of type **AbEntry** that match the search request.
AbEntry	Container for one user or distribution list that match the search request.
ArrayOfAttribute	Sequence of elements of type **Attribute** that are the set of requested attributes for the matched entry.
Attribute	Container for the name and value of one attribute.
ArrayOfstring	Sequence of elements of type **string**.
AbEntryResponse.ResponseMetadata	Contains the response status and optional message text associated with the response status.

3.1.4.2.3.1 SearchAbEntry

The **SearchAbEntry** complex type is the overall container of the information that is sent in the SOAP request to search the address book. The schema of the request body within the SOAP envelope is as follows:

```
<s:element name="SearchAbEntry">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="AbEntryRequest"
type="tns:AbEntryRequest" />
        </s:sequence>
    </s:complexType>
</s:element>
```

SearchAbEntry.AbEntryRequest: This element is of type **AbEntryRequest** and contains the actual request. There MUST be exactly one such element present in each **SearchAbEntry** request.

3.1.4.2.3.2 AbEntryRequest

The **AbEntryRequest** complex type is the container for the information that describes the type of search being requested and the parameters to the search. It MUST contain one and only one of **BasicSearch**, **ChangeSearch**, or **OrgSearch** elements. It MUST contain one and only one **Metadata** element. The schema for this complex type is as follows:

```
<s:complexType name="AbEntryRequest">
  <s:sequence>
    <s:element minOccurs="0" name="BasicSearch" type="tns:AbEntryRequest.BasicSearchQuery" />
    <s:element minOccurs="0" name="ChangeSearch"
type="tns:ArrayOfAbEntryRequest.ChangeSearchQuery" />
    <s:element minOccurs="0" name="Metadata" type="tns:AbEntryRequest.SearchMetadata" />
    <s:element minOccurs="0" name="OrgSearch" type="tns:AbEntryRequest.OrgSearchQuery" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntryRequest.BasicSearch: The **BasicSearch** element describes a basic search request, either an exact match or a prefix match, of a value against one or more attribute values associated with users and distribution lists in the **Global Address List (GAL)**. The type of the **BasicSearch** element is **AbEntryRequest.BasicSearchQuery**.

AbEntryRequest.ChangeSearch: The **ChangeSearch** element describes a search for changes. The type of the **ChangeSearch** element is **ArrayOfAbEntryRequest.ChangeSearchQuery**.

AbEntryRequest.Metadata: The **Metadata** element contains information that modifies how a search is performed. The type of the **Metadata** element is **AbEntryRequest.SearchMetadata**.

AbEntryRequest.OrgSearch: The **OrgSearch** element describes a search to retrieve the organizational structure for a given user or distribution list. The type of the **OrgSearch** element is **AbEntryRequest.OrgSearchQuery**.

3.1.4.2.3.3 AbEntryRequest.BasicSearchQuery

The **AbEntryRequest.BasicSearchQuery** complex type is the container for the information that describes a basic search request. The schema for this complex type is as follows:

```
<s:complexType name="AbEntryRequest.BasicSearchQuery">
  <s:sequence>
    <s:element minOccurs="0" name="SearchList" type="s:string" />
    <s:element minOccurs="0" name="Value" type="s:string" />
```

```
      <s:element name="Verb" type="tns:SearchVerb" />
    </s:sequence>
  </s:complexType>
```

The following elements are contained within this complex type:

AbEntryRequest.BasicSearchQuery.SearchList: A string type that contains a comma-separated list of attribute names whose values are to be searched against. Invalid or empty names are ignored. If the entire element is missing, empty, or specifies only invalid attribute names, all attributes are searched. For a description of attribute names, see section 3.1.1.

AbEntryRequest.BasicSearchQuery.Value: A string type that contains the value to search for. This value is compared against each of the values associated with the attributes named by the **SearchList** element. The comparison is case and accent insensitive.

AbEntryRequest.BasicSearchQuery.Verb: The type of search, either exact match or prefix match. There MUST be exactly one **Verb** element contained in the **BasicSearch** element. The **Verb** element is of simple type **SearchVerb**. The **SearchVerb** type is described in section 3.1.4.2.4.1.

3.1.4.2.3.4 ArrayOfAbEntryRequest.ChangeSearchQuery

The **ArrayOfAbEntryRequest.ChangeSearchQuery** complex type consists of a sequence of elements of type **AbEntryRequest.ChangeSearchQuery**. This type contains information about each user being searched for. The schema for this complex type is as follows:

```
  <s:complexType name="ArrayOfAbEntryRequest.ChangeSearchQuery">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="100" name="AbEntryRequest.ChangeSearchQuery"
  type="tns:AbEntryRequest.ChangeSearchQuery" />
    </s:sequence>
  </s:complexType>
```

The following element is contained within this complex type:

ArrayOfAbEntryRequest.**ChangeSearchQuery.AbEntryRequest.ChangeSearchQuery**: Contains information about each change search query. This element is of type **AbEntryRequest.ChangeSearchQuery**. This element can occur from zero to 100 times in the **ArrayOfAbEntryRequest.ChangeSearchQuery** element.

3.1.4.2.3.5 AbEntryRequest.ChangeSearchQuery

The **AbEntryRequest.ChangeSearchQuery** complex type contains information about each user being searched for. The schema for this complex type is as follows:

```
  <s:complexType name="AbEntryRequest.ChangeSearchQuery">
    <s:sequence>
      <s:element minOccurs="0" name="AbEntryHash" type="s:string" />
      <s:element minOccurs="0" name="PhotoHash" type="s:string" />
      <s:element minOccurs="0" name="SearchOn" type="s:string" />
      <s:element minOccurs="0" name="Value" type="s:string" />
    </s:sequence>
  </s:complexType>
```

The following elements are contained within this complex type:

AbEntryRequest.ChangeSearchQuery.AbEntryHash: Identifies the client's view of a particular contact. If the search request matches a user or distribution list and the hash of the matching entry matches this field, the server MUST NOT return any attribute values for this entry. If the hash values do not match, the server MUST return all attribute values requested for the matched entry.

AbEntryRequest.ChangeSearchQuery.PhotoHash: Identifies the client's view of a particular contact's photo. If the search request matches a user or distribution list and the hash of the matching entry's photo matches this field, the server MUST NOT return any photo attribute data for this entry. If the hash values do not match, the server MUST return all requested photo attributes for the matched entry.

AbEntryRequest.ChangeSearchQuery.SearchOn: A string type that contains a comma-separated list of attribute names whose values are to be searched against. Invalid or empty names are ignored. If the entire element is missing, empty, or specifies only invalid attribute names, all attributes MUST be searched. For a description of attribute names, see section 3.1.1.

AbEntryRequest.ChangeSearchQuery.Value: A string type that contains the value to search for. This value is compared against each of the values associated with the attributes named by the **ReturnList** element as specified in section 3.1.4.2.3.6. The search is an exact match for all characters and is case and accent insensitive.

3.1.4.2.3.6 AbEntryRequest.SearchMetadata

The **AbEntryRequest.SearchMetadata** complex type contains information that modifies how a search is performed. The schema for this complex type is as follows:

```
<s:complexType name="AbEntryRequest.SearchMetadata">
  <s:sequence>
    <s:element name="FromDialPad" type="s:boolean" />
    <s:element name="MaxResultNum" type="s:unsignedInt" />
    <s:element minOccurs="0" name="ReturnList" type="s:string" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntryRequest.SearchMetadata.FromDialPad: A Boolean type which, if "true", tells the server it MUST treat all search strings in this request as numeric strings from a dial pad and match against the possible letters and digits on each dial key. For example, "12" matches "1a", "1ả", "1b", "1c", or "12", ignoring case. The server MUST map directory data as follows when creating the dial pad indexes to search against:

- If a character is one of "*.?!,@'-_:;()&"~¹º¿¡%£$¥¤¢+×/\[]=<>§", map it to an asterisk ("*").
- If a character is a letter (a through z) ignoring case, map to the correct number (abc maps to "2", def maps to "3", and so forth).
- If a character is a number (0 through 9), map it as is.
- Otherwise drop the character from the index.

AbEntryRequest.SearchMetadata.MaxResultNum: An integer type that specifies the maximum number of results to return for this search query. The default value is "20". The server MUST NOT return more entries than the number specified by this field.

AbEntryRequest.SearchMetadata.ReturnList: A string type that contains a comma-separated list of attribute names whose values are to be returned for each matching entry. Invalid or empty names MUST be ignored. If the entire element is empty or specifies only invalid attribute names, all attributes MUST be returned for each matching entry. If the element is not specified, the server SHOULD return "InvalidArgumentError". For a description of attribute names, see section 3.1.1.

3.1.4.2.3.7 AbEntryRequest.OrgSearchQuery

The **AbEntryRequest.OrgSearchQuery** complex type contains information that describes an organizational search request. The schema for this complex type is as follows:

```
<s:complexType name="AbEntryRequest.OrgSearchQuery">
  <s:sequence>
    <s:element minOccurs="0" name="EntryId" type="s:string" />
    <s:element minOccurs="0" name="OrgHash" type="s:string" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntryRequest.OrgSearchQuery.EntryId: A string type that uniquely identifies the user or distribution list whose organizational structure is being queried.

AbEntryRequest.OrgSearchQuery.OrgHash: A string that represents the client's view of the entry's organizational structure. If the server hash value matched, the server MUST NOT return any information for this search. If the server hash value does not match, the server MUST return all the entries in the matching entries organizational structure.

3.1.4.2.3.8 SearchAbEntryResponse

The **SearchAbEntryResponse** complex type is the overall container in the response to the **SearchAbEntry** request. **SearchAbEntryResponse** encapsulates the results of the operation to search the Global Address List (GAL). It contains one **SearchAbEntryResult** element of type **AbEntryResponse**. The schema for this complex type within the SOAP envelope is as follows:

```
<s:element name="SearchAbEntryResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" name="SearchAbEntryResult" type="tns:AbEntryResponse" />
    </s:sequence>
  </s:complexType>
</s:element>
```

The following element MUST be contained in the **SearchAbEntryResponse** complex type:

SearchAbEntryResponse.SearchAbEntryResult: This element is of type **AbEntryResponse** and contains information about the results of the search operation against the Global Address List (GAL).

3.1.4.2.3.9 AbEntryResponse

The **AbEntryResponse** complex type is the container for the results of the search request. It contains one **Items** element of type **ArrayOfAbEntry**. The schema for this complex type is as follows:

```
<s:complexType name="AbEntryResponse">
  <s:sequence>
    <s:element minOccurs="0" name="Items" type="tns:ArrayOfAbEntry" />
    <s:element minOccurs="0" name="Metadata" type="tns:AbEntryResponse.ResponseMetadata" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntryResponse.Items: Information about each of the address book entries that match the **SearchAbEntry** request. There MUST be exactly one **Items** element in the **AbEntryResponse** element. The type of the **Items** element is complex type **ArrayOfAbEntry**.

AbEntryResponse.Metadata: Information about the overall success or failure of the search request. There MUST be exactly one **Metadata** element in the **SearchAbEntryResult** element. The type of the **Metadata** element is complex type **AbEntryResponse.ResponseMetadata** as specified in section 3.1.4.2.3.15.

3.1.4.2.3.10 ArrayOfAbEntry

The **ArrayOfAbEntry** complex type consists of a sequence of elements of type **AbEntry**. This type contains Global Address List (GAL) information about each user or distribution list that matched the search request. The schema for this complex type is as follows:

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

The following element is contained within this complex type:

ArrayOfAbEntry.AbEntry: Information about one Global Address List (GAL)user or distribution list. The type of the **AbEntry** element is complex type **AbEntry**.

3.1.4.2.3.11 AbEntry

The **AbEntry** complex type is the container for information about one user or distribution list from the Global Address List (GAL). The schema for this complex type is as follows:

```
<s:complexType name="AbEntry">
  <s:sequence>
    <s:element minOccurs="0" name="Attributes" type="tns:ArrayOfAttribute" />
    <s:element minOccurs="0" name="EntryId" type="s:string" />
    <s:element minOccurs="0" name="Position" type="s:int" />
    <s:element minOccurs="0" name="SourceNetwork" type="s:string" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntry.Attributes: The attributes from the Global Address List (GAL)for each user or distribution list that matched the search request. There MUST be exactly one **Attributes** element in the **AbEntry** element. The type of the **Attributes** element is complex type **ArrayOfAttribute**.

AbEntry.**EntryId**: A string type that uniquely identifies the user or distribution list in the Global Address List (GAL).

AbEntry.**Position**: An integer type that contains a value that identifies the position of the user or distribution list in the organization hierarchy. For organizational search requests, the server MUST set this element to an integer that gives the relative position of this entry to the entry requested in the organizational search. A value of zero ("0") MUST be returned for the entry requested and the peers of that entry. A value of "-1" MUST be returned for the direct reports of the entry requested. A value of "1" MUST be returned for the manager of the entry requested. A value of "2" MUST be returned for the manager’s manager.

For distribution lists, the server MUST return only one or two entries for an organizational search:

- The distribution list entry, where the **Position** element MUST be zero ("0").
- If present, the owner of the distribution list, where the **Position** element MUST be "1".

AbEntry.**SourceNetwork**[<8>](#): A string type that indicates the source directory from where this user entry was extracted. It is set to "**SameEnterprise**" for all AbEntry objects returned by this API.

3.1.4.2.3.12 ArrayOfAttribute

The **ArrayofAttribute** complex type consists of a sequence of elements of type **Attribute**. This type contains attribute information about a single user or distribution list. The schema for this complex type is as follows:

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

The following element is contained within this complex type:

ArrayOfAttribute.**Attribute**: The **Attribute** element contains information about one attribute for one user or distribution list in the Global Address List (GAL). The type of the **Attribute** element is the complex type **Attribute**.

3.1.4.2.3.13 Attribute

The **Attribute** complex type is the container for information about one attribute. The schema for this complex type is as follows:

```
<s:complexType name="Attribute">
  <s:sequence>
    <s:element minOccurs="0" name="Name" type="s:string" />
    <s:element minOccurs="0" name="Value" type="s:string" />
    <s:element minOccurs="0" name="Values" type="tns:ArrayOfstring" />
  </s:sequence>
</s:complexType>
```

The container MUST contain one **Attribute.Value** element or one **Attribute.Values** element. The container MUST NOT contain both **Attribute.Value** and **Attribute.Values** elements.

Attribute.**Name (String)**: The name of the attribute.

Attribute.**Value (String)**: The value of the attribute. The server MUST use this element when an attribute has a single value.

Attribute.**Values**: The values for a multi-value attribute. The server MUST use this element when an attribute has more than one value. The type of the **Values** element is complex type **ArrayOfstring**.

3.1.4.2.3.14 ArrayOfstring

The **ArrayOfstring** complex type consists of a sequence of elements of type string. The schema for this complex type is as follows:

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

```
    </s:complexType>
```

The following element is contained within this complex type:

ArrayOfString.**string**: A string type that contains an attribute value.

3.1.4.2.3.15 AbEntryResponse.ResponseMetadata

The **AbEntryResponse.ResponseMetadata** complex type is the container for metadata about the response to a search request. The schema for this complex type is as follows:

```
    <s:complexType name="AbEntryResponse.ResponseMetadata">
      <s:sequence>
        <s:element minOccurs="0" name="MessageText" type="s:string" />
        <s:element minOccurs="0" name="ResponseCode" type="tns:SearchResponseState" />
      </s:sequence>
    </s:complexType>
```

The following elements are contained within this complex type:

AbEntryResponse.**ResponseMetadata**.**MessageText (String)**: Unlocalized text associated with the **ResponseCode** that gives further information.

AbEntryResponse.**ResponseMetadata**.**ResponseCode**: The result of the **SearchAbEntry** request and MUST be present. There MUST be exactly one of these elements in the **ResponseMetadata** element. This element is of simple type **SearchResponseState**. The **SearchResponseState** type is described in section 3.1.4.2.4.2.

3.1.4.2.4 Simple Types

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

Simple type	Description
SearchVerb	Describes the type of **BasicSearch** to perform, either an exact match or a prefix match.
SearchResponseState	Completion status of the attempted **SearchAbEntry** request.

3.1.4.2.4.1 SearchVerb

SearchVerb is a simple type that is an enumeration. The schema for this type is as follows:

```
    <s:simpleType name="SearchVerb">
      <s:restriction base="s:string">
        <s:enumeration value="Equals" />
        <s:enumeration value="BeginsWith" />
      </s:restriction>
    </s:simpleType>
```

The following table lists the meaning of each enumerated value.

Value	Meaning
Equals	The entire search string matches one of the attributes being searched against.
BeginsWith	The search string partially matches the beginning of one of the attributes being searched against.

3.1.4.2.4.2 SearchResponseState

SearchResponseState is a simple type that is an enumeration. The schema for this type is as follows:

```
<s:simpleType name="SearchResponseState">
  <s:restriction base="s:string">
    <s:enumeration value="Succeeded" />
    <s:enumeration value="NoEntryFound" />
    <s:enumeration value="InternalError" />
    <s:enumeration value="InvalidArgumentError" />
    <s:enumeration value="DatabaseError" />
    <s:enumeration value="CorruptionEntryError" />
  </s:restriction>
</s:simpleType>
```

The following table lists the meaning of each enumerated value:

Value	Meaning
Succeeded	The server MUST return this string in the **ResponseCode** element when a search request has successfully matched one or more entries in the Global Address List (GAL).
NoEntryFound	The server MUST return this string in the **ResponseCode** element when a search request failed to match any entries in the Global Address List (GAL).
InternalError	The server SHOULD return this string in the **ResponseCode** element when it is unable to complete a request because of an unexpected internal error.
InvalidArgumentError	The server MUST return this string in the **ResponseCode** element when one or more arguments are invalid.
DatabaseError	The server SHOULD return this string in the **ResponseCode** element when it encounters an error when attempting to access the Global Address List (GAL).
CorruptionEntryError	The server SHOULD return this string in the **ResponseCode** element when it matches an entry in the Global Address List (GAL)but the entry is corrupted.

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 SearchSkypeDirectory

The following excerpt from this protocol's WSDL specifies the messages that constitute this operation<9>:

```
<wsdl:operation name="SearchSkypeDirectory">
  <wsdl:input message="tns:SearchSkypeDirectorySoapIn" />
  <wsdl:output message="tns:SearchSkypeDirectorySoapOut" />
</wsdl:operation>
```

When user action triggers a request to search the Skype public directory, a TCP connection MUST be made to the server and SSL MUST be negotiated. The address of the server that makes the TCP connection MUST be obtained through a separate channel, such as SIP. After successful SSL negotiation, a SOAP HTTP request containing a **SearchSkypeDirectorySoapIn** message MUST be constructed with a SOAP body containing the **SearchSkypeDirectory** element.

On receiving a **SearchSkypeDirectory** request, the server SHOULD query the Skype graph search service to search for the information specified in the request. After conducting the requested search, the server MUST construct the **SearchSkypeDirectorySoapOut** message, containing the **SearchSkypeDirectoryResponse** element, and it MUST send the message in the SOAP HTTP response, which is a 2xx response to a SOAP HTTP request. In case of errors, the response message MUST specify the error and the type of error that was encountered by the server.

3.1.4.3.1 Messages

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

Message	Description
SearchSkypeDirectorySoapIn	Request from the client to search the Skype public directory.
SearchSkypeDirectorySoapOut	Response from the server after it executes a request to search the Skype public directory.

3.1.4.3.1.1 SearchSkypeDirectorySoapIn

The **SearchSkypeDirectorySoapIn** SOAP message is a request that is sent from the client. For example, a **SearchSkypeDirectorySoapIn** message is created as a result of a user action to search for a user in the Skype public directory . The request information MUST be captured in the **SearchSkypeDirectory** element in the SOAP body of the message. The **SearchSkypeDirectory** element is specified in section 3.1.4.3.3.1.

3.1.4.3.1.2 SearchSkypeDirectorySoapOut

This SOAP message is a response that is sent by the server after it executes a request to search the Skype public directory. This message contains the result of the search on the server. The result is represented in the **SearchSkypeDirectoryResponse** element, which MUST be in the SOAP body of the SOAP message. The **SearchSkypeDirectoryResponse** element is specified in section 3.1.4.3.3.3.

3.1.4.3.2 Elements

All elements in the WSDL are contained in complex types and specified in section 3.1.4.3.3

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
SearchSkypeDirectory	The overall container of the request to search the Skype public directory.
SkypeDirectorySearchRequest	Search request container that MUST contain **SearchQuery**, **RequestId**, and **MaxResultNum** fields**.**
SearchSkypeDirectoryResponse	The overall container of the response to the Skype public directory search request. **SearchSkypeDirectoryResponse** contains the response status and, if the response is successful, the users that match the request.
AbEntryResponse	Container for the results of the search.
ArrayOfAbEntry	Sequence of elements of type **AbEntry** that match the search request.
AbEntry	Container for one user that match the search request.
ArrayOfAttribute	Sequence of elements of type **Attribute** that are the set of available attributes for the matched entry.
Attribute	Container for the name and value of one attribute.
ArrayOfstring	Sequence of elements of type **string**.
AbEntryResponse.ResponseMetadata	Contains the response status and optional message text associated with the response status.

3.1.4.3.3.1 SearchSkypeDirectory

The **SearchSkypeDirectory** complex type is the overall container of the information that is sent in the SOAP request to search the Skype public directory. The schema of the request body within the SOAP envelope is as follows:

```
<s:element name="SearchSkypeDirectory">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="skypeDirectorySearchRequest"
type="tns:SkypeDirectorySearchRequest" />
        </s:sequence>
```

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

SearchSkypeDirectory.skypeDirectorySearchRequest: This element is of type **SkypeDirectorySearchRequest** and contains the actual request. There MUST be exactly one such element present in each **SearchSkypeDirectory** request.

3.1.4.3.3.2 SkypeDirectorySearchRequest

The **SkypeDirectorySearchRequest** complex type is the container for the information that describes the type of search being requested and the parameters to the search. The schema for this complex type is as follows:

```
    <s:complexType name="SkypeDirectorySearchRequest">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="MaxResultNum" type="xs:unsignedInt" />
        <s:element minOccurs="1" maxOccurs="1" name="RequestId" type="xs:string" />
        <s:element minOccurs="1" maxOccurs="1" name="SearchQuery" type="xs:string" />
        <s:element minOccurs="1" maxOccurs="1" name="ExecuteAnonymously" type="xs:boolean" />
      </s:sequence>
    </s:complexType>
```

The following elements are contained within this complex type:

SkypeDirectorySearchRequest.**MaxResultNum**: The **MaxResultNum** field MUST contain a positive integer value that indicates maximum number of search results to be returned.

SkypeDirectorySearchRequest.**RequestId**: The **RequestId** field MUST contain a string value that uniquely identifies the current search request.

SkypeDirectorySearchRequest.SearchQuery: The **SearchQuery** field MUST contain a string value that will be used as a search query while searching for users in the Skype public directory.

SkypeDirectorySearchRequest.ExecuteAnonymously: This is a Boolean flag. When it is set to true, user’s encrypted identity is not sent to Skype public directory search service.

3.1.4.3.3.3 SearchSkypeDirectoryResponse

The **SearchSkypeDirectoryResponse** complex type is the overall container in the response to the **SearchSkypeDirectory** request. **SearchSkypeDirectoryResponse** encapsulates the results of the operation to search the Skype public directory . It contains one **SearchSkypeDirectoryResult** element of type **AbEntryResponse**. The schema for this complex type within the SOAP envelope is as follows:

```
    <s:element name="SearchSkypeDirectoryResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" name="SearchSkypeDirectoryResult" type="tns:AbEntryResponse"
    />
        </s:sequence>
      </s:complexType>
    </s:element>
```

The following element MUST be contained in the **SearchSkypeDirectoryResponse** complex type:

SearchSkypeDirectoryResponse.SearchSkypeDirectoryResult: This element is of type **AbEntryResponse** and contains information about the results of the search operation against the Skype public directory.

3.1.4.3.3.4 AbEntryResponse

The **AbEntryResponse** complex type is the container for the results of the Skype public directory search request. It contains one **Items** element of type **ArrayOfAbEntry**. The schema for this complex type is as follows:

```
<s:complexType name="AbEntryResponse">
  <s:sequence>
    <s:element minOccurs="0" name="Items" type="tns:ArrayOfAbEntry" />
    <s:element minOccurs="0" name="Metadata" type="tns:AbEntryResponse.ResponseMetadata" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntryResponse.Items: Information about each of the users in Skype public directory that match the **SearchSkypeDirectory** request. There MUST be exactly one **Items** element in the **AbEntryResponse** element. The type of the **Items** element is complex type **ArrayOfAbEntry**.

AbEntryResponse.Metadata: Information about the overall success or failure of the search request. There MUST be exactly one **Metadata** element in the **SearchSkypeDirectoryResult** element. The type of the **Metadata** element is complex type **AbEntryResponse.ResponseMetadata** as specified in section 3.1.4.2.3.15

3.1.4.3.3.5 ArrayOfAbEntry

The **ArrayOfAbEntry** complex type consists of a sequence of elements of type **AbEntry**. This type contains information about each user in the Skype public directory that matched the search request. The schema for this complex type is as follows:

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

The following element is contained within this complex type:

ArrayOfAbEntry.AbEntry: Information about one user in Skype public directory. The type of the **AbEntry** element is complex type **AbEntry**.

3.1.4.3.3.6 AbEntry

The **AbEntry** complex type is the container for information about one user from the Skype public directory. The schema for this complex type is as follows:

```
<s:complexType name="AbEntry">
  <s:sequence>
    <s:element minOccurs="0" name="Attributes" type="tns:ArrayOfAttribute" />
    <s:element minOccurs="0" name="EntryId" type="s:string" />
    <s:element minOccurs="0" name="Position" type="s:int" />
    <s:element minOccurs="0" name="SourceNetwork" type="s:string" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

AbEntry.Attributes: The attributes from the Skype public directory for each user or distribution list that matched the search request. There MUST be exactly one **Attributes** element in the **AbEntry** element. The type of the **Attributes** element is complex type **ArrayOfAttribute**.

AbEntry.**EntryId**: This field is not being used at present.

AbEntry.**Position**: This field is not being used at present.

AbEntry.SourceNetwork: A string type that indicates the source directory from where this user entry was extracted. It is set to "**SkypePublic**" for all AbEntry objects returned by this API.

3.1.4.3.3.7 ArrayOfAttribute

The **ArrayofAttribute** complex type consists of a sequence of elements of type **Attribute**. This type contains attribute information about a single user. The schema for this complex type is as follows:

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

The following element is contained within this complex type:

ArrayOfAttribute.**Attribute**: The **Attribute** element contains information about one attribute for one user in the Skype public directory. The type of the **Attribute** element is the complex type **Attribute**.

3.1.4.3.3.8 Attribute

The **Attribute** complex type is the container for information about one attribute. The schema for this complex type is as follows:

```
<s:complexType name="Attribute">
  <s:sequence>
    <s:element minOccurs="0" name="Name" type="s:string" />
    <s:element minOccurs="0" name="Value" type="s:string" />
    <s:element minOccurs="0" name="Values" type="tns:ArrayOfstring" />
  </s:sequence>
</s:complexType>
```

The container MUST contain one **Attribute.Value** element or one **Attribute.Values** element. The container MUST NOT contain both **Attribute.Value** and **Attribute.Values** elements.

Attribute.**Name (String)**: The name of the attribute.

Attribute.**Value (String)**: The value of the attribute. The server MUST use this element when an attribute has a single value.

Attribute.**Values**: The values for a multi-value attribute. The server MUST use this element when an attribute has more than one value. The type of the **Values** element is complex type **ArrayOfstring**.

3.1.4.3.3.9 ArrayOfstring

The **ArrayOfstring** complex type consists of a sequence of elements of type string. The schema for this complex type is as follows:

```
<s:complexType name="ArrayOfstring">
  <s:sequence>
```

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

The following element is contained within this complex type:

ArrayOfString.**string**: A string type that contains an attribute value.

3.1.4.3.3.10 AbEntryResponse.ResponseMetadata

The **AbEntryResponse.ResponseMetadata** complex type is the container for metadata about the response to a search request. The schema for this complex type is as follows:

```
  <s:complexType name="AbEntryResponse.ResponseMetadata">
    <s:sequence>
      <s:element minOccurs="0" name="MessageText" type="s:string" />
      <s:element minOccurs="0" name="ResponseCode" type="tns:SearchResponseState" />
    </s:sequence>
  </s:complexType>
```

The following elements are contained within this complex type:

AbEntryResponse.**ResponseMetadata**.**MessageText (String)**: Unlocalized text associated with the **ResponseCode** that gives further information.

AbEntryResponse.**ResponseMetadata**.**ResponseCode**: The result of the **SearchSkypeDirectory** request and MUST be present. There MUST be exactly one of these elements in the **ResponseMetadata** element. This element is of simple type **SearchResponseState**. The **SearchResponseState** type is described in section 3.1.4.2.4.2.

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
SearchResponseState	Completion status of the attempted **SearchAbEntry** request.

3.1.4.3.4.1 SearchResponseState

SearchResponseState is a simple type that is an enumeration. The schema for this type is as follows:

```
  <s:simpleType name="SearchResponseState">
    <s:restriction base="s:string">
      <s:enumeration value="Succeeded" />
      <s:enumeration value="NoEntryFound" />
      <s:enumeration value="InternalError" />
      <s:enumeration value="InvalidArgumentError" />
      <s:enumeration value="DatabaseError" />
      <s:enumeration value="CorruptionEntryError" />
    </s:restriction>
  </s:simpleType>
```

The following table lists the meaning of each enumerated value:

Value	Meaning
Succeeded	The server MUST return this string in the **ResponseCode** element when a search request has successfully matched one or more entries in the Skype public directory.
NoEntryFound	The server MUST return this string in the **ResponseCode** element when a search request failed to match any entries in the Skype public directory.
InternalError	The server SHOULD return this string in the **ResponseCode** element when it is unable to complete a request because of an unexpected internal error.
InvalidArgumentError	The server MUST return this string in the **ResponseCode** element when one or more arguments are invalid.
DatabaseError	This value is not used at present.
CorruptionEntryError	This value is not used at present.

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 ProvideSkypeSearchFeedback

The following excerpt from this protocol's WSDL specifies the messages that constitute this operation<10>:

```
<wsdl:operation name="ProvideSkypeSearchFeedback">
  <wsdl:input message="tns:ProvideSkypeSearchFeedbackSoapIn" />
  <wsdl:output message="tns:ProvideSkypeSearchFeedbackSoapOut" />
</wsdl:operation>
```

When a user interacts with one or more results from Skype public directory search operation, it could trigger a request to provide feedback on the Skype public directory search operation. When user action triggers this operation, a TCP connection MUST be made to the server and SSL MUST be negotiated. The address of the server that makes the TCP connection MUST be obtained through a separate channel, such as SIP. After successful SSL negotiation, a SOAP HTTP request containing a **ProvideSkypeSearchFeedbackSoapIn** message MUST be constructed with a SOAP body containing the **ProvideSkypeSearchFeedback** element.

On receiving a **ProvideSkypeSearchFeedback** request, the server SHOULD forward the feedback information specified in the request to Skype public directory search service. After submitting the specified feedback data, the server MUST construct the **ProvideSkypeSearchFeedbackSoapOut** message, containing the **ProvideSkypeSearchFeedbackResponse** element, and it MUST send the message in the SOAP HTTP response, which is a 2xx response to a SOAP HTTP request. In case of errors, the response message MUST specify the error and the type of error that was encountered by the server.

3.1.4.4.1 Messages

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

Message	Description
ProvideSkypeSearchFeedbackSoapIn	Request from the client to submit feedback on Skype public directory search results.
ProvideSkypeSearchFeedbackSoapOut	Response from the server after it submits the feedback data to Skype public directory search service.

3.1.4.4.1.1 ProvideSkypeSearchFeedbackSoapIn

The **ProvideSkypeSearchFeedbackSoapIn** SOAP message is a request that is sent from the client. For example, a **ProvideSkypeSearchFeedbackSoapIn** message is created as a result of a user action to provide feedback on Skype public directory search results. The request information MUST be captured in the **ProvideSkypeSearchFeedback** element in the SOAP body of the message. The **ProvideSkypeSearchFeedback** element is specified in section 3.1.4.4.3.1

3.1.4.4.1.2 ProvideSkypeSearchFeedbackSoapOut

This SOAP message is a response that is sent by the server after it executes a request to provide feedback on Skype public directory search results. This message contains the result of the request to provide feedback to Skype public directory search service. The result is represented in the **ProvideSkypeSearchFeedbackResponse** element, which MUST be in the SOAP body of the SOAP message. The **ProvideSkypeSearchFeedbackResponse** element is specified in section 3.1.4.4.3.3.

3.1.4.4.2 Elements

All elements in the WSDL are contained in complex types are specified in section 3.1.4.4.3.

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
PovideSkypeSearchFeedback	The overall container of the request that provides feedback on Skype public directory search results.
SkypeSearchFeedbackRequest	Skype search feedback request container.
PovideSkypeSearchFeedbackResponse	The overall container of the response to the request that provides feedback data.
SkypeSearchFeedbackResponse	**SkypeSearchFeedbackResponse** contains the response status and, if the response is successful.
ArrayOfstring	Sequence of elements of type **string**.

3.1.4.4.3.1 ProvideSkypeSearchFeedback

The **ProvideSkypeSearchFeedback** complex type is the overall container of the information that is sent in the SOAP request to submit feedback data on Skype public directory search results. The schema of the request body within the SOAP envelope is as follows:

```
<s:element name="ProvideSkypeSearchFeedback">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="skypeSearchFeedbackRequest"
type="tns: SkypeSearchFeedbackRequest" />
        </s:sequence>
    </s:complexType>
</s:element>
```

ProvideSkypeSearchFeedback.skypeSearchFeedbackRequest: This element is of type **SkypeSearchFeedbackRequest** and contains the actual request. There MUST be exactly one such element present in each **ProvideSkypeSearchFeedback** request.

3.1.4.4.3.2 SkypeSearchFeedbackRequest

The **SkypeSearchFeedbackRequest** complex type is the container for the information that provides feedback on the results of Skype public directory search operation. The schema for this complex type is as follows:

```
<s:complexType name="SkypeSearchFeedbackRequest">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="RequestId" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="SkypeIds" type="tns:ArrayOfstring " />
    <s:element minOccurs="1" maxOccurs="1" name="ExecuteAnonymously" type="xs:boolean" />
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

SkypeSearchFeedbackRequest.**RequestId**: The **RequestId** element **MUST** be specified and it should match the request Id of the original Skype public directory search operation for which feedback is being sent.

SkypeSearchFeedbackRequest.**SkypeIds**: The **SkypeIds** element is an array of strings that contains list of one or more Skype Ids from within the list of search results that were selected by the user as good results.

SkypeSearchFeedbackRequest.ExecuteAnonymously: This is a Boolean flag. When it is set to true, user’s encrypted identity is not sent to Skype public directory search service.

3.1.4.4.3.3 ProvideSkypeSearchFeedbackResponse

The **ProvideSkypeSearchFeedbackResponse** complex type is the overall container in the response to the **ProvideSkypeSearchFeedback** request. **ProvideSkypeSearchFeedbackResponse** encapsulates the results of the operation to submit feedback on Skype public directory search results. It contains one **ProvideSkypeSearchFeedbackResults** element of type **SkypeSearchFeedbackResponse**. The schema for this complex type within the SOAP envelope is as follows:

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

```
      <s:element minOccurs="0" name="ProvideSkypeSearchFeedbackResult"
type="tns:AbEntryResponse" />
    </s:sequence>
  </s:complexType>
</s:element>
```

The following element MUST be contained in the **ProvideSkypeSearchFeedbackResponse** complex type:

ProvideSkypeSearchFeedbackResponse.ProvideSkypeSearchFeedbackResult: This element is of type **SkypeSearchFeedbackResponse** and contains information about the results of the operation to submit feedback about Skype public directory search results.

3.1.4.4.3.4 SkypeSearchFeedbackResponse

The **SkypeSearchFeedbackResponse** complex type is the container for the results of the operation to submit feedback on Skype public directory search results. The schema for this complex type is as follows:

```
<s:complexType name="SkypeSearchFeedbackResponse">
  <s:sequence>
    <s:element minOccurs="0" name="MessageText" type="tns:string" />
    <s:element minOccurs="0" name="ResponseCode" type="tns:SkypeSearchFeedbackResponseCode"
/>
  </s:sequence>
</s:complexType>
```

The following elements are contained within this complex type:

SkypeSearchFeedbackResponse.MessageText: Un-localized text associated with the **ResponseCode** that gives further information.

SkypeSearchFeedbackResponse.ResponseCode: The result of the **ProvideSkypeSearchFeedback** request and MUST be present. This element is of simple type **SkypeSearchFeedbackResponseCode**. The **SkypeSearchFeedbackResponseCode** type is described in section 3.1.4.4.4.1.

3.1.4.4.3.5 ArrayOfString

The **ArrayOfstring** complex type consists of a sequence of elements of type string. The schema for this complex type is as follows:

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

The following element is contained within this complex type:

ArrayOfString.string: A string type that contains an attribute value.

3.1.4.4.4 Simple Types

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

Simple type	Description
SkypeSearchFeedbackResponseCode	Completion status of the attempted **ProvideSkypeSearchFeedback** request.

3.1.4.4.4.1 SkypeSearchFeedbackResponseCode

SkypeSearchFeedbackResponseCode is a simple type that is an enumeration. The schema for this type is as follows:

```
<s:simpleType name="SkypeSearchFeedbackResponseCode">
  <s:restriction base="s:string">
    <s:enumeration value="Succeeded" />
    <s:enumeration value="InternalError" />
    <s:enumeration value="InvalidArgumentError" />
  </s:restriction>
</s:simpleType>
```

The following table lists the meaning of each enumerated value:

Value	Meaning
Succeeded	The server MUST return this string in the **ResponseCode** element when the feedback information is successfully submitted to Skype public directory search service.
InternalError	The server SHOULD return this string in the **ResponseCode** element when it is unable to complete a request because of an unexpected internal error.
InvalidArgumentError	The server MUST return this string in the **ResponseCode** element when one or more arguments are invalid.

3.1.4.4.5 Attributes

None

3.1.4.4.6 Groups

None

3.1.4.4.7 Attribute Groups

None

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Successful Distribution List Expansion Request and Response

The following example is an **ExpandDistributionList Request**. This request is sent from a client to the server as a SOAP HTTP request.

```
<soap:Body>
  <ExpandDistributionList xmlns="DistributionListExpander">
    <groupMailAddress>sales@contoso.com</groupMailAddress>
  </ExpandDistributionList>
</soap:Body>
```

This request results in the following successful SOAP HTTP response.

```
<soap:Body>
  <ExpandDistributionListResponse xmlns="DistributionListExpander">
    <ExpandDistributionListResult>
      <ResponseStatus>Success</ResponseStatus>
      <Users>
        <ActiveDirectoryObjectInfo>
          <displayName>Don Hall</displayName>
          <mail>don@contoso.com</mail>
          <mailNickname>don</mailNickname>
          <sipUri>sip:don@contoso.com</sipUri>
        </ActiveDirectoryObjectInfo>
        <ActiveDirectoryObjectInfo>
          <displayName>Eran Harel</displayName>
          <mail>Eran@contoso.com</mail>
          <mailNickname>Eran</mailNickname>
          <sipUri>sip:Eran@contoso.com</sipUri>
        </ActiveDirectoryObjectInfo>
      </Users>
      <NestedGroups />
    </ExpandDistributionListResult>
  </ExpandDistributionListResponse>
</soap:Body>
```

4.2 Successful Distribution List Expansion Request and Response with Nested Groups

The following is an example of a distribution list expansion in a SOAP HTTP request. Note that this request is identical to the request shown in the example in section 4.1.

```
<soap:Body>
  <ExpandDistributionList xmlns="DistributionListExpander">
    <groupMailAddress>sales@contoso.com</groupMailAddress>
  </ExpandDistributionList>
</soap:Body>
```

This request results in the following successful SOAP HTTP response with nested groups. Note that some of the members of the distribution list are themselves distribution lists that are listed under the **NestedGroups** element.

```
<soap:Body>
  <ExpandDistributionListResponse xmlns="DistributionListExpander">
    <ExpandDistributionListResult>
      <ResponseStatus>Success</ResponseStatus>
      <Users>
        <ActiveDirectoryObjectInfo>
          <displayName>Don Hall</displayName>
          <mail>Don@contoso.com</mail>
          <mailNickname>Don</mailNickname>
          <sipUri>sip:Don@contoso.com</sipUri>
        </ActiveDirectoryObjectInfo>
        <ActiveDirectoryObjectInfo>
          <displayName>Eran Harel</displayName>
          <mail>Eran@contoso.com</mail>
          <mailNickname>Eran</mailNickname>
          <sipUri>sip:Eran@contoso.com</sipUri>
        </ActiveDirectoryObjectInfo>
        <ActiveDirectoryObjectInfo>
          <displayName>Joe Healy</displayName>
          <mail>Joe@contoso.com</mail>
          <mailNickname>Joe</mailNickname>
          <sipUri>sip:Joe@contoso.com</sipUri>
        </ActiveDirectoryObjectInfo>
      </Users>
      <NestedGroups>
        <ActiveDirectoryObjectInfo>
          <displayName>Marketing</displayName>
          <mail>marketing@contoso.com</mail>
          <mailNickname>marketing</mailNickname>
        </ActiveDirectoryObjectInfo>
        <ActiveDirectoryObjectInfo>
          <displayName>Accounting</displayName>
          <mail>accounting@contoso.com</mail>
          <mailNickname>accounting</mailNickname>
        </ActiveDirectoryObjectInfo>
      </NestedGroups>
    </ExpandDistributionListResult>
  </ExpandDistributionListResponse>
</soap:Body>
```

4.3 Unsuccessful Distribution List Expansion Request and Response

The following is an example of a distribution list expansion SOAP HTTP request. Note that this request is identical to the requests in the examples in section 4.1 and section 4.2.

```
<soap:Body>
  <ExpandDistributionList xmlns="DistributionListExpander">
    <groupMailAddress>sales@contoso.com</groupMailAddress>
  </ExpandDistributionList>
</soap:Body>
```

This request results in the following unsuccessful SOAP HTTP response. Note that the **ResponseStatus** is not "success" in this case but is set to the reason that the request failed. In this particular case, the request failed because the count of the number of members in the distribution list exceeded a configured limit.

```
<soap:Body>
  <ExpandDistributionListResponse xmlns="DistributionListExpander">
    <ExpandDistributionListResult>
      <ResponseStatus>MemberCountLimitExceeded</ResponseStatus>
      <Users />
      <NestedGroups />
    </ExpandDistributionListResult>
  </ExpandDistributionListResponse>
</soap:Body>
```

4.4 Successful Basic Search Request and Response using exact match

This request is an example of a search using an exact match against the **displayName** attribute value.

```
    <soap:Body>
      <AbEntryRequest xmlns="DistributionListExpander">
        <BasicSearch>
          <SearchList>displayName</SearchList>
          <Value>TZ_tester</Value>
          <Verb>Equals</Verb>
        </BasicSearch>
        <Metadata>
          <ReturnList>displayName</ReturnList>
        </Metadata>
      </AbEntryRequest>
    </soap:Body>
```

This request results in the following successful SOAP HTTP response with the matching entry and the requested attributes.

```
    <soap:Body>
      <AbEntryResponse xmlns="DistributionListExpander">
        <Items>
          <AbEntry>
            <Attributes>
              <Attribute>
                <Name>displayname</Name>
                <Value>TZ_tester</Value>
              </Attribute>
            </Attributes>
            <EntryId>79d7099e-a85d-499d-a2c6-32b002937cf4</EntryId>
            <Position>0</Position>
          </AbEntry>
        </Items>
        <Metadata>
          <ResponseCode>Succeeded</ResponseCode>
        </Metadata>
      </AbEntryResponse>
    </soap:Body>
```

4.5 Successful Basic Search Request and Response using prefix match

This request is an example of a search prefix match against the **displayName** attribute value.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <BasicSearch>
      <SearchList>displayName</SearchList>
      <Value>TZ_</Value>
      <Verb>BeginsWith</Verb>
    </BasicSearch>
    <Metadata>
      <ReturnList>displayName</ReturnList>
    </Metadata>
  </AbEntryRequest>
</soap:Body>
```

This request results in the following successful SOAP HTTP response with the matching entry and the requested attributes.

```
<soap:Body>
  <AbEntryResponse xmlns="DistributionListExpander">
    <Items>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_tester</Value>
          </Attribute>
        </Attributes>
        <EntryId>79d7099e-a85d-499d-a2c6-32b002937cf4</EntryId>
        <Position>0</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_Grp_manager1</Value>
          </Attribute>
        </Attributes>
        <EntryId>fb0b875d-c25f-4d3d-bfdb-718f4d398dcc</EntryId>
        <Position>0</Position>
      </AbEntry>
    </Items>
    <Metadata>
      <ResponseCode>Succeeded</ResponseCode>
    </Metadata>
  </AbEntryResponse>
</soap:Body>
```

4.6 Unsuccessful Basic Search Request and Response

This request is an example of a search using an exact match against the **displayName** attribute value that fails to find a match.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <BasicSearch>
      <SearchList>displayName</SearchList>
      <Value>NotSuchName</Value>
      <Verb>Equals</Verb>
    </BasicSearch>
    <Metadata>
```

```
        <ReturnList>displayName</ReturnList>
      </Metadata>
    </AbEntryRequest>
  </soap:Body>
```

This request results in the following unsuccessful SOAP HTTP response.

```
  <soap:Body>
    <AbEntryResponse xmlns="DistributionListExpander">
      <Items />
      <Metadata>
        <ResponseCode>NoEntryFound</ResponseCode>
      </Metadata>
    </AbEntryResponse>
  </soap:Body>
```

4.7 Successful Change Search Request and Response

This request is an example of a change search for two users.

```
  <soap:Body>
    <AbEntryRequest xmlns="DistributionListExpander">
      <ChangeSearch>
        <AbEntryRequest.ChangeSearchQuery>
          <SearchOn>displayName</SearchOn>
          <Value>vt1_user0</Value>
        </AbEntryRequest.ChangeSearchQuery>
        <AbEntryRequest.ChangeSearchQuery>
          <SearchOn>displayName</SearchOn>
          <Value>vt1_user1</Value>
        </AbEntryRequest.ChangeSearchQuery>
      </ChangeSearch>
      <Metadata>
        <ReturnList>displayName,AbEntryHash</ReturnList>
      </Metadata>
    </AbEntryRequest>
  </soap:Body>
```

This request results in the following successful SOAP HTTP response with the matched entries and the requested attributes.

```
  <soap:Body>
    <AbEntryResponse xmlns="DistributionListExpander">
      <Items>
        <AbEntry>
          <Attributes>
            <Attribute>
              <Name>displayname</Name>
              <Value>vt1_user0</Value>
            </Attribute>
            <Attribute>
              <Name>AbEntryHash</Name>
              <Value>865038ec-92ed-7db2-a52a-4591a92a4829</Value>
            </Attribute>
          </Attributes>
          <EntryId>dc913538-677f-4fef-8c80-1e2615bfde61</EntryId>
          <Position>0</Position>
        </AbEntry>
        <AbEntry>
          <Attributes>
            <Attribute>
              <Name>displayname</Name>
              <Value>vt1_user1</Value>
            </Attribute>
            <Attribute>
              <Name>AbEntryHash</Name>
              <Value>038ec-92ed-7db2-a52a-4591a92a4829</Value>
            </Attribute>
          </Attributes>
          <EntryId>e92d7790-3668-4974-88ee-3d34c5d24e76</EntryId>
          <Position>0</Position>
        </AbEntry>
      </Items>
      <Metadata>
        <ResponseCode>Succeeded</ResponseCode>
      </Metadata>
    </AbEntryResponse>
  </soap:Body>
```

4.8 Successful Change Search Request and Response using entry hash

This request is an example of a change search for two users with an **AbEntryHash** specified for each.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <ChangeSearch>
      <AbEntryRequest.ChangeSearchQuery>
        <SearchOn>displayName</SearchOn>
        <Value>vt1_user0</Value>
        <AbEntryHash>865038ec-92ed-7db2-a52a-4591a92a4829</AbEntryHash>
      </AbEntryRequest.ChangeSearchQuery>
      <AbEntryRequest.ChangeSearchQuery>
        <SearchOn>displayName</SearchOn>
        <Value>vt1_user1</Value>
        <AbEntryHash>038ec-92ed-7db2-a52a-4591a92a4829</AbEntryHash>
      </AbEntryRequest.ChangeSearchQuery>
    </ChangeSearch>
    <Metadata>
      <ReturnList>displayName,title,AbEntryHash</ReturnList>
    </Metadata>
  </AbEntryRequest>
</soap:Body>
```

This request results in the following successful SOAP HTTP response with the matched entries. Note that only the second matched entry contains attribute values as it is the only one whose **AbEntryHash** value has changed.

```
<soap:Body>
  <AbEntryResponse xmlns="DistributionListExpander">
    <Items>
      <AbEntry>
        <Attributes />
        <EntryId>dc913538-677f-4fef-8c80-1e2615bfde61</EntryId>
        <Position>0</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>vt1_user1</Value>
          </Attribute>
          <Attribute>
            <Name>title</Name>
            <Value>New Title</Value>
          </Attribute>
          <Attribute>
            <Name>AbEntryHash</Name>
            <Value>b724c2ef-9774-417a-83d8-c2869eeadf08</Value>
          </Attribute>
        </Attributes>
        <EntryId>e92d7790-3668-4974-88ee-3d34c5d24e76</EntryId>
        <Position>0</Position>
      </AbEntry>
    </Items>
    <Metadata>
      <ResponseCode>Succeeded</ResponseCode>
    </Metadata>
  </AbEntryResponse>
</soap:Body>
```

4.9 Successful Change Search Request and Response using entry and photo hash

This request is an example of a change search for two users with an **AbEntryHash** specified for each.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <ChangeSearch>
      <AbEntryRequest.ChangeSearchQuery>
        <SearchOn>displayName</SearchOn>
        <Value>vt1_user0</Value>
        <PhotoHash>ee3979ae-fde8-4493-bf2b-c2171b17e928</PhotoHash>
      </AbEntryRequest.ChangeSearchQuery>
      <AbEntryRequest.ChangeSearchQuery>
        <SearchOn>displayName</SearchOn>
        <Value>vt1_user1</Value>
        <PhotoHash>4aa71c2f-64ab-4cc0-b969-78d7c90a32b3</PhotoHash>
      </AbEntryRequest.ChangeSearchQuery>
    </ChangeSearch>
    <Metadata>
      <ReturnList>displayName,PhotoRelPath,PhotoHash</ReturnList>
    </Metadata>
  </AbEntryRequest>
</soap:Body>
```

This request results in the following successful SOAP HTTP response with the matched entries. Note that only the second matched entry contains attribute values because it is the only one whose **PhotoHash** value has changed.

```
<soap:Body>
  <AbEntryResponse xmlns="DistributionListExpander">
    <Items>
      <AbEntry>
        <Attributes />
        <EntryId>dc913538-677f-4fef-8c80-1e2615bfde61</EntryId>
        <Position>0</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>vt1_user1</Value>
          </Attribute>
          <Attribute>
            <Name>PhotoRelPath</Name>
            <Value> e92d7790-3668-4974-88ee-3d34c5d24e76.86b20bbc-648a-44af-97ba-
505e9cf7427d.photo</Value>
          </Attribute>
          <Attribute>
```

```
            <Name>PhotoHash</Name>
            <Value>86b20bbc-648a-44af-97ba-505e9cf7427d</Value>
          </Attribute>
        </Attributes>
        <EntryId>e92d7790-3668-4974-88ee-3d34c5d24e76</EntryId>
        <Position>0</Position>
      </AbEntry>
    </Items>
    <Metadata>
      <ResponseCode>Succeeded</ResponseCode>
    </Metadata>
  </AbEntryResponse>
</soap:Body>
```

4.10 Successful Change Search Request and Response with not found entries

This request is an example of a change search for two users.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <ChangeSearch>
      <AbEntryRequest.ChangeSearchQuery>
        <SearchOn>displayName</SearchOn>
        <Value>vt1_user0</Value>
      </AbEntryRequest.ChangeSearchQuery>
      <AbEntryRequest.ChangeSearchQuery>
        <SearchOn>displayName</SearchOn>
        <Value>vt1_user1</Value>
      </AbEntryRequest.ChangeSearchQuery>
    </ChangeSearch>
    <Metadata>
      <ReturnList>displayName,AbEntryHash</ReturnList>
    </Metadata>
  </AbEntryRequest>
</soap:Body>
```

This request results in the following successful SOAP HTTP response with one matched entry and the requested attributes. The second entry was not found, so it is not returned. The response status is "succeeded", however, even if one or more entries are not found.

```
<soap:Body>
  <AbEntryResponse xmlns="DistributionListExpander">
    <Items>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>vt1_user0</Value>
          </Attribute>
          <Attribute>
            <Name>AbEntryHash</Name>
            <Value>865038ec-92ed-7db2-a52a-4591a92a4829</Value>
          </Attribute>
        </Attributes>
        <EntryId>dc913538-677f-4fef-8c80-1e2615bfde61</EntryId>
        <Position>0</Position>
      </AbEntry>
    </Items>
    <Metadata>
      <ResponseCode>Succeeded</ResponseCode>
    </Metadata>
  </AbEntryResponse>
</soap:Body>
```

4.11 Successful Organization Search Request and Response

This is an example of a search for the organizational structure for a user.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <Metadata>
      <ReturnList>displayName,OrgHash</ReturnList>
    </Metadata>
    <OrgSearch>
      <EntryId>44961af4-de8e-4d85-9c0b-d2e0a88da584</EntryId>
    </OrgSearch>
  </AbEntryRequest>
</soap:Body>
```

This request results in the following successful SOAP HTTP response with nine entries that represent two direct reports, in the position -1, three peers, in the position 0, and four managers in the management chain, in the positions of 1, 2, 3 and 4. Note that the **OrgHash** attribute value is only returned for the entry whose **EntryId** was passed in the request.

```
<soap:Body>
  <AbEntryResponse xmlns="DistributionListExpander">
    <Items>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU13</Value>
          </Attribute>
        </Attributes>
        <EntryId>28a6e7b3-9c97-4592-88cc-0c4805bdb68d</EntryId>
        <Position>-1</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU14</Value>
          </Attribute>
        </Attributes>
        <EntryId>969e1ded-7af5-491e-8040-ea4f4a9192c6</EntryId>
        <Position>-1</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU12</Value>
          </Attribute>
        </Attributes>
        <EntryId>a6853350-d8a6-4a1e-bae7-332b9580ccc2</EntryId>
        <Position>0</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU11</Value>
          </Attribute>
        </Attributes>
        <EntryId>03844533-b8b8-4f88-9903-7167759240a1</EntryId>
        <Position>0</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
```

```
            <Name>displayname</Name>
            <Value>TZ_orgSearchU10</Value>
          </Attribute>
          <Attribute>
            <Name>OrgHash</Name>
            <Value>1b320d60-7419-4c97-8c61-055b2e77ce8d</Value>
          </Attribute>
        </Attributes>
        <EntryId>44961af4-de8e-4d85-9c0b-d2e0a88da584</EntryId>
        <Position>0</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU6</Value>
          </Attribute>
        </Attributes>
        <EntryId>9d5d05e5-70a3-4291-9200-b6a2b433770e</EntryId>
        <Position>1</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU7</Value>
          </Attribute>
        </Attributes>
        <EntryId>8f73e70b-4619-45c5-a120-260fb35d755a</EntryId>
        <Position>2</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU8</Value>
          </Attribute>
        </Attributes>
        <EntryId>42a79101-9017-41c1-a264-cb64f05f980e</EntryId>
        <Position>3</Position>
      </AbEntry>
      <AbEntry>
        <Attributes>
          <Attribute>
            <Name>displayname</Name>
            <Value>TZ_orgSearchU9</Value>
          </Attribute>
        </Attributes>
        <EntryId>2b6b4bf8-84d5-4158-bb99-876e32088e1e</EntryId>
        <Position>4</Position>
      </AbEntry>
    </Items>
    <Metadata>
      <ResponseCode>Succeeded</ResponseCode>
    </Metadata>
  </AbEntryResponse>
</soap:Body>
```

4.12 Successful Organization Search Request and Response using organization hash

This is an example of a successful organization search using an **OrgHash** value returned by a previous search.

```
<soap:Body>
  <AbEntryRequest xmlns="DistributionListExpander">
    <Metadata>
```

```
        <ReturnList>displayName,OrgHash</ReturnList>
      </Metadata>
      <OrgSearch>
        <EntryId>44961af4-de8e-4d85-9c0b-d2e0a88da584</EntryId>
        <OrgHash>1b320d60-7419-4c97-8c61-055b2e77ce8d</OrgHash>
      </OrgSearch>
    </AbEntryRequest>
  </soap:Body>
```

This request results in a successful SOAP HTTP response with no entries because the **OrgHash** value has not changed for the requested entry.

```
  <soap:Body>
    <AbEntryResponse xmlns="DistributionListExpander">
      <Items />
      <Metadata>
        <ResponseCode>Succeeded</ResponseCode>
      </Metadata>
    </AbEntryResponse>
  </soap:Body>
```

4.13 Unsuccessful Organization Search Request and Response

This is an example of an unsuccessful organization search using an unknown **EntryId** value.

```
  <soap:Body>
    <AbEntryRequest xmlns="DistributionListExpander">
      <Metadata>
        <ReturnList>displayName,OrgHash</ReturnList>
      </Metadata>
      <OrgSearch>
        <EntryId>fba32d62-5b6c-4f54-a853-e5d8968ee601</EntryId>
        <OrgHash>1b320d60-7419-4c97-8c61-055b2e77ce8d</OrgHash>
      </OrgSearch>
    </AbEntryRequest>
  </soap:Body>
```

This request results in an unsuccessful SOAP HTTP response with no entries because the **EntryId** is not known. The **OrgHash** value is ignored if **EntryId** is not found.

```
  <soap:Body>
    <AbEntryResponse xmlns="DistributionListExpander">
      <Items />.
      <Metadata>
        <ResponseCode>NoEntryFound</ResponseCode>
      </Metadata>
    </AbEntryResponse>
  </soap:Body>
```

4.14 Successful Skype Directory Search Request and Response

This is an example of a successful Skype directory search request and response.

```
  <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <SearchSkypeDirectory xmlns="DistributionListExpander">
        <skypeDirectorySearchRequest>
```

```
            <MaxResultNum>50</MaxResultNum>
            <RequestId>9ea968b5-f692-4eb3-86fc-eb16fb7c24a0</RequestId>
            <SearchQuery>test</SearchQuery>
          </skypeDirectorySearchRequest>
        </SearchSkypeDirectory>
      </s:Body>
    </s:Envelope>
```

This request results in the following successful **SOAP HTTP** response with two user entries.

```
    <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
      <s:Body>
        <SearchSkypeDirectoryResponse xmlns="DistributionListExpander">
          <SearchSkypeDirectoryResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <Items>
              <AbEntry>
                <Attributes>
                  <Attribute>
                    <Name>skypeId</Name>
                    <Value>test.user1</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>name</Name>
                    <Value>test user1</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>avatarUrl</Name>
                    <Value>https://api.skype.com/users/test.user1/profile/avatar</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>city</Name>
                    <Value>Redmond</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>state</Name>
                    <Value>Washington</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>country</Name>
                    <Value>USA</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                </Attributes>
                <EntryId i:nil="true"/>
                <Position>0</Position>
                <SourceNetwork>SkypePublic</SourceNetwork>
              </AbEntry>
              <AbEntry>
                <Attributes>
                  <Attribute>
                    <Name>skypeId</Name>
                    <Value>test.user2</Value>
                    <Values i:nil="true" 
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
```

```
                    <Name>name</Name>
                    <Value>test user2</Value>
                    <Values i:nil="true"
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>avatarUrl</Name>
                    <Value>https://api.skype.com/users/test.user2/profile/avatar</Value>
                    <Values i:nil="true"
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>city</Name>
                    <Value>Redmond</Value>
                    <Values i:nil="true"
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>state</Name>
                    <Value>WA</Value>
                    <Values i:nil="true"
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                  <Attribute>
                    <Name>country</Name>
                    <Value>USA</Value>
                    <Values i:nil="true"
    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                  </Attribute>
                </Attributes>
                <EntryId i:nil="true"/>
                <Position>0</Position>
                <SourceNetwork>SkypePublic</SourceNetwork>
              </AbEntry>
            </Items>
            <Metadata>
              <MessageText i:nil="true"/>
              <ResponseCode>Succeeded</ResponseCode>
            </Metadata>
          </SearchSkypeDirectoryResult>
        </SearchSkypeDirectoryResponse>


  </s:Body>
</s:Envelope>
```

4.15 Unsuccessful Skype Directory Search Request and Response

This is an example of an unsuccessful Skype Directory Search request.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SearchSkypeDirectory xmlns="DistributionListExpander">
      <skypeDirectorySearchRequest>
        <MaxResultNum>50</MaxResultNum>
        <RequestId xsi:nil="true"/>
        <SearchQuery>a</SearchQuery>
      </skypeDirectorySearchRequest>
    </SearchSkypeDirectory>
  </s:Body>
</s:Envelope>
```

In this request, one of the required fields (RequestId) is not specified. As a result, server responds back with an error (InvalueArgumentError).

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SearchSkypeDirectoryResponse xmlns="DistributionListExpander">
      <SearchSkypeDirectoryResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Items i:nil="true"/>
        <Metadata>
          <MessageText>
            Some of the required fields are missing in the request.
          </MessageText>
          <ResponseCode>InvalidArgumentError</ResponseCode>
        </Metadata>
      </SearchSkypeDirectoryResult>
    </SearchSkypeDirectoryResponse>
  </s:Body>
</s:Envelope>
```

4.16 Successful Provide Skype Search Feedback Request and Response

This is an example of a successful Provide Skype Search Feedback request and response.

```
  <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <ProvideSkypeSearchFeedback xmlns="DistributionListExpander">
        <skypeSearchFeedbackRequest>
          <RequestId>9ea968b5-f692-4eb3-86fc-eb16fb7c24a0</RequestId>
          <SkypeIds>
            <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">test</string>
          </SkypeIds>
        </skypeSearchFeedbackRequest>
      </ProvideSkypeSearchFeedback>
    </s:Body>
  </s:Envelope>
```

This request results in the following successful SOAP HTTP response with two user entries.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ProvideSkypeSearchFeedbackResponse xmlns="DistributionListExpander">
      <ProvideSkypeSearchFeedbackResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
        <MessageText i:nil="true"/>
        <ResponseCode>Succeeded</ResponseCode>
      </ProvideSkypeSearchFeedbackResult>
    </ProvideSkypeSearchFeedbackResponse>
  </s:Body>


</s:Envelope>
```

4.17 Unsuccessful Provide Skype Search Feedback Request and Response

This is an example of an unsuccessful request to Provide Skype search feedback.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
```

```
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <ProvideSkypeSearchFeedback xmlns="DistributionListExpander">
      <skypeSearchFeedbackRequest>
        <RequestId xsi:nil="true"/>
        <SkypeIds>
          <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">abc</string>
        </SkypeIds>
      </skypeSearchFeedbackRequest>
    </ProvideSkypeSearchFeedback>
  </s:Body>
</s:Envelope>
```

Since one of the required fields (RequestId) is not specified, server responds back with an error (InvalueArgumentError)

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ProvideSkypeSearchFeedbackResponse xmlns="DistributionListExpander">
      <ProvideSkypeSearchFeedbackResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
        <MessageText>
          Some of the required fields are missing in the request.
        </MessageText>
        <ResponseCode>InvalidArgumentError</ResponseCode>
      </ProvideSkypeSearchFeedbackResult>
    </ProvideSkypeSearchFeedbackResponse>
  </s:Body>
</s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

The distribution list protocol allows HTTP connections only over SSL. Users are authenticated using **Kerberos** v5 and **NT LAN Manager (NTLM) Authentication Protocol** authentication methods, as described in [MS-NLMP]. Clients can also be authenticated using Kerberos and NTLM HTTP authentication, as described in [RFC4559].

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.

```
<wsdl:definitions name="Service" targetNamespace="DistributionListExpander"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="DistributionListExpander"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <wsdl:types>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/">
      <xs:element name="anyType" nillable="true" type="xs:anyType"/>
      <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
      <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
      <xs:element name="boolean" nillable="true" type="xs:boolean"/>
      <xs:element name="byte" nillable="true" type="xs:byte"/>
      <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
      <xs:element name="decimal" nillable="true" type="xs:decimal"/>
      <xs:element name="double" nillable="true" type="xs:double"/>
      <xs:element name="float" nillable="true" type="xs:float"/>
      <xs:element name="int" nillable="true" type="xs:int"/>
      <xs:element name="long" nillable="true" type="xs:long"/>
      <xs:element name="QName" nillable="true" type="xs:QName"/>
      <xs:element name="short" nillable="true" type="xs:short"/>
      <xs:element name="string" nillable="true" type="xs:string"/>
      <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
      <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
      <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
      <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
      <xs:element name="char" nillable="true" type="tns:char"/>
      <xs:simpleType name="char">
        <xs:restriction base="xs:int"/>
      </xs:simpleType>
      <xs:element name="duration" nillable="true" type="tns:duration"/>
      <xs:simpleType name="duration">
        <xs:restriction base="xs:duration">
          <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
          <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
          <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:element name="guid" nillable="true" type="tns:guid"/>
      <xs:simpleType name="guid">
        <xs:restriction base="xs:string">
          <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-
F]{12}"/>
```

```
    </xs:restriction>
   </xs:simpleType>
   <xs:attribute name="FactoryType" type="xs:QName"/>
   <xs:attribute name="Id" type="xs:ID"/>
   <xs:attribute name="Ref" type="xs:IDREF"/>
  </xs:schema>
  <xs:schema elementFormDefault="qualified" targetNamespace="DistributionListExpander"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
   <xs:import namespace="http://microsoft.com/wsdl/types/"/>
   <xs:simpleType name="SearchVerb">
    <xs:restriction base="xs:string">
     <xs:enumeration value="Equals"/>
     <xs:enumeration value="BeginsWith"/>
    </xs:restriction>
   </xs:simpleType>
   <xs:element name="SearchVerb" nillable="true" type="tns:SearchVerb"/>
   <xs:simpleType name="SearchResponseState">
    <xs:restriction base="xs:string">
     <xs:enumeration value="Succeeded"/>
     <xs:enumeration value="NoEntryFound"/>
     <xs:enumeration value="InternalError"/>
     <xs:enumeration value="InvalidArgumentError"/>
     <xs:enumeration value="DatabaseError"/>
     <xs:enumeration value="CorruptionEntryError"/>
    </xs:restriction>
   </xs:simpleType>
   <xs:element name="SearchResponseState" nillable="true" type="tns:SearchResponseState"/>
   <xs:element name="SearchAbEntry">
    <xs:complexType>
     <xs:sequence>
      <xs:element minOccurs="0" name="AbEntryRequest" nillable="true"
type="tns:AbEntryRequest"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:complexType name="AbEntryRequest">
    <xs:sequence>
     <xs:element minOccurs="0" name="BasicSearch" nillable="true"
type="tns:AbEntryRequest.BasicSearchQuery"/>
     <xs:element minOccurs="0" name="ChangeSearch" nillable="true"
type="tns:ArrayOfAbEntryRequest.ChangeSearchQuery"/>
     <xs:element minOccurs="0" name="Metadata" nillable="true"
type="tns:AbEntryRequest.SearchMetadata"/>
     <xs:element minOccurs="0" name="OrgSearch" nillable="true"
type="tns:AbEntryRequest.OrgSearchQuery"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="AbEntryRequest" nillable="true" type="tns:AbEntryRequest"/>
   <xs:complexType name="AbEntryRequest.BasicSearchQuery">
    <xs:sequence>
     <xs:element minOccurs="0" name="SearchList" nillable="true" type="xs:string"/>
     <xs:element minOccurs="0" name="Value" nillable="true" type="xs:string"/>
     <xs:element name="Verb" type="tns:SearchVerb"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="AbEntryRequest.BasicSearchQuery" nillable="true"
type="tns:AbEntryRequest.BasicSearchQuery"/>
```

```
   <xs:complexType name="ArrayOfAbEntryRequest.ChangeSearchQuery">
    <xs:sequence>
     <xs:element minOccurs="0" maxOccurs="unbounded"
name="AbEntryRequest.ChangeSearchQuery" nillable="true"
type="tns:AbEntryRequest.ChangeSearchQuery"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="ArrayOfAbEntryRequest.ChangeSearchQuery" nillable="true"
type="tns:ArrayOfAbEntryRequest.ChangeSearchQuery"/>
   <xs:complexType name="AbEntryRequest.ChangeSearchQuery">
    <xs:sequence>
     <xs:element minOccurs="0" name="AbEntryHash" nillable="true" type="xs:string"/>
     <xs:element minOccurs="0" name="PhotoHash" nillable="true" type="xs:string"/>
     <xs:element minOccurs="0" name="SearchOn" nillable="true" type="xs:string"/>
     <xs:element minOccurs="0" name="Value" nillable="true" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="AbEntryRequest.ChangeSearchQuery" nillable="true"
type="tns:AbEntryRequest.ChangeSearchQuery"/>
   <xs:complexType name="AbEntryRequest.SearchMetadata">
    <xs:sequence>
     <xs:element name="FromDialPad" nillable="true" type="xs:boolean"/>
     <xs:element name="MaxResultNum" nillable="true" type="xs:unsignedInt"/>
     <xs:element minOccurs="0" name="ReturnList" nillable="true" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="AbEntryRequest.SearchMetadata" nillable="true"
type="tns:AbEntryRequest.SearchMetadata"/>
   <xs:complexType name="AbEntryRequest.OrgSearchQuery">
    <xs:sequence>
     <xs:element minOccurs="0" name="EntryId" nillable="true" type="xs:string"/>
     <xs:element minOccurs="0" name="OrgHash" nillable="true" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="AbEntryRequest.OrgSearchQuery" nillable="true"
type="tns:AbEntryRequest.OrgSearchQuery"/>
   <xs:element name="SearchAbEntryResponse">
    <xs:complexType>
     <xs:sequence>
      <xs:element minOccurs="0" name="SearchAbEntryResult" nillable="true"
type="tns:AbEntryResponse"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:complexType name="AbEntryResponse">
    <xs:sequence>
     <xs:element minOccurs="0" name="Items" nillable="true" type="tns:ArrayOfAbEntry"/>
     <xs:element minOccurs="0" name="Metadata" nillable="true"
type="tns:AbEntryResponse.ResponseMetadata"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="AbEntryResponse" nillable="true" type="tns:AbEntryResponse"/>
   <xs:complexType name="ArrayOfAbEntry">
    <xs:sequence>
     <xs:element minOccurs="0" maxOccurs="unbounded" name="AbEntry" nillable="true"
type="tns:AbEntry"/>
    </xs:sequence>
   </xs:complexType>
```

```
    <xs:element name="ArrayOfAbEntry" nillable="true" type="tns:ArrayOfAbEntry"/>
    <xs:complexType name="AbEntry">
     <xs:sequence>
      <xs:element minOccurs="0" name="Attributes" nillable="true" type="tns:ArrayOfAttribute"/>
      <xs:element minOccurs="0" name="EntryId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Position" type="xs:int"/>
      <xs:element minOccurs="0" name="SourceNetwork" nillable="true" type="xs:string"/>
     </xs:sequence>
    </xs:complexType>
    <xs:element name="AbEntry" nillable="true" type="tns:AbEntry"/>
    <xs:complexType name="ArrayOfAttribute">
     <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Attribute" nillable="true"
type="tns:Attribute"/>
     </xs:sequence>
    </xs:complexType>
    <xs:element name="ArrayOfAttribute" nillable="true" type="tns:ArrayOfAttribute"/>
    <xs:complexType name="Attribute">
     <xs:sequence>
      <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Value" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Values" nillable="true" type="q1:ArrayOfstring"
xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
     </xs:sequence>
    </xs:complexType>
    <xs:element name="Attribute" nillable="true" type="tns:Attribute"/>
    <xs:complexType name="AbEntryResponse.ResponseMetadata">
     <xs:sequence>
      <xs:element minOccurs="0" name="MessageText" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="ResponseCode" type="tns:SearchResponseState"/>
     </xs:sequence>
    </xs:complexType>
    <xs:element name="AbEntryResponse.ResponseMetadata" nillable="true"
type="tns:AbEntryResponse.ResponseMetadata"/>
    <xs:element name="ExpandDistributionList">
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs="0" maxOccurs="1" name="groupMailAddress" type="xs:string"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="ExpandDistributionListResponse">
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs="0" maxOccurs="1" name="ExpandDistributionListResult"
type="tns:DlxGroup"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:complexType name="DlxGroup">
     <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="ResponseStatus"
type="tns:ResponseState"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Users"
type="tns:ArrayOfActiveDirectoryObjectInfo"/>
      <xs:element minOccurs="0" maxOccurs="1" name="NestedGroups"
type="tns:ArrayOfActiveDirectoryObjectInfo"/>
     </xs:sequence>
```

```
      </xs:complexType>
      <xs:simpleType name="ResponseState">
        <xs:restriction base="xs:string">
          <xs:enumeration value="Invalid"/>
          <xs:enumeration value="Success"/>
          <xs:enumeration value="MemberCountLimitExceeded"/>
          <xs:enumeration value="NotAuthorized"/>
          <xs:enumeration value="NotFound"/>
          <xs:enumeration value="SimultaneousRequestLimitExceeded"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="ArrayOfActiveDirectoryObjectInfo">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="ActiveDirectoryObjectInfo"
type="tns:ActiveDirectoryObjectInfo"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ActiveDirectoryObjectInfo">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="displayName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="mail" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="mailNickname" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="sipUri" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="SearchSkypeDirectory">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="skypeDirectorySearchRequest" nillable="true"
type="tns:SkypeDirectorySearchRequest"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="SkypeDirectorySearchRequest">
        <xs:sequence>
          <xs:element name="ExecuteAnonymously" type="xs:boolean"/>
          <xs:element name="MaxResultNum" nillable="true" type="xs:unsignedInt"/>
          <xs:element name="RequestId" nillable="true" type="xs:string"/>
          <xs:element name="SearchQuery" nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="SkypeDirectorySearchRequest" nillable="true"
type="tns:SkypeDirectorySearchRequest"/>
      <xs:element name="SearchSkypeDirectoryResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="SearchSkypeDirectoryResult" nillable="true"
type="tns:AbEntryResponse"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ProvideSkypeSearchFeedback">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="skypeSearchFeedbackRequest" nillable="true"
type="tns:SkypeSearchFeedbackRequest"/>
          </xs:sequence>
        </xs:complexType>
```

```
   </xs:element>
   <xs:complexType name="SkypeSearchFeedbackRequest">
    <xs:sequence>
     <xs:element name="ExecuteAnonymously" type="xs:boolean"/>
     <xs:element name="RequestId" nillable="true" type="xs:string"/>
     <xs:element name="SkypeIds" nillable="true" type="q3:ArrayOfstring"
xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="SkypeSearchFeedbackRequest" nillable="true"
type="tns:SkypeSearchFeedbackRequest"/>
   <xs:element name="ProvideSkypeSearchFeedbackResponse">
    <xs:complexType>
     <xs:sequence>
      <xs:element minOccurs="0" name="ProvideSkypeSearchFeedbackResult" nillable="true"
type="tns:SkypeSearchFeedbackResponse"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:complexType name="SkypeSearchFeedbackResponse">
    <xs:sequence>
     <xs:element name="MessageText" nillable="true" type="xs:string"/>
     <xs:element name="ResponseCode" type="tns:SkypeSearchFeedbackResponseCode"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="SkypeSearchFeedbackResponse" nillable="true"
type="tns:SkypeSearchFeedbackResponse"/>
   <xs:simpleType name="SkypeSearchFeedbackResponseCode">
    <xs:restriction base="xs:string">
     <xs:enumeration value="Succeeded"/>
     <xs:enumeration value="InternalError"/>
     <xs:enumeration value="InvalidArgumentError"/>
    </xs:restriction>
   </xs:simpleType>
   <xs:element name="SkypeSearchFeedbackResponseCode" nillable="true"
type="tns:SkypeSearchFeedbackResponseCode"/>
  </xs:schema>
  <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <xs:complexType name="ArrayOfstring">
    <xs:sequence>
     <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
   <xs:element name="ArrayOfstring" nillable="true" type="tns:ArrayOfstring"/>
  </xs:schema>
  <xs:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://microsoft.com/wsdl/types/">
   <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
     <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{12}"/>
    </xs:restriction>
   </xs:simpleType>
```

```
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="SearchAbEntrySoapIn">
    <wsdl:part name="parameters" element="tns:SearchAbEntry"/>
  </wsdl:message>
  <wsdl:message name="SearchAbEntrySoapOut">
    <wsdl:part name="parameters" element="tns:SearchAbEntryResponse"/>
  </wsdl:message>
  <wsdl:message name="ExpandDistributionListSoapIn">
    <wsdl:part name="parameters" element="tns:ExpandDistributionList"/>
  </wsdl:message>
  <wsdl:message name="ExpandDistributionListSoapOut">
    <wsdl:part name="parameters" element="tns:ExpandDistributionListResponse"/>
  </wsdl:message>
  <wsdl:message name="SearchSkypeDirectorySoapIn">
    <wsdl:part name="parameters" element="tns:SearchSkypeDirectory"/>
  </wsdl:message>
  <wsdl:message name="SearchSkypeDirectorySoapOut">
    <wsdl:part name="parameters" element="tns:SearchSkypeDirectoryResponse"/>
  </wsdl:message>
  <wsdl:message name="ProvideSkypeSearchFeedbackSoapIn">
    <wsdl:part name="parameters" element="tns:ProvideSkypeSearchFeedback"/>
  </wsdl:message>
  <wsdl:message name="ProvideSkypeSearchFeedbackSoapOut">
    <wsdl:part name="parameters" element="tns:ProvideSkypeSearchFeedbackResponse"/>
  </wsdl:message>
  <wsdl:portType name="IAddressBook">
    <wsdl:operation name="SearchAbEntry">
      <wsdl:input wsaw:Action="DistributionListExpander/IAddressBook/SearchAbEntry" 
message="tns:SearchAbEntrySoapIn"/>
      <wsdl:output wsaw:Action="DistributionListExpander/IAddressBook/SearchAbEntryResponse" 
message="tns:SearchAbEntrySoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="ExpandDistributionList">
      <wsdl:input wsaw:Action="DistributionListExpander/ExpandDistributionList" 
message="tns:ExpandDistributionListSoapIn"/>
      <wsdl:output 
wsaw:Action="DistributionListExpander/IAddressBook/ExpandDistributionListResponse" 
message="tns:ExpandDistributionListSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="SearchSkypeDirectory">
      <wsdl:input wsaw:Action="DistributionListExpander/IAddressBook/SearchSkypeDirectory" 
message="tns:SearchSkypeDirectorySoapIn"/>
      <wsdl:output 
wsaw:Action="DistributionListExpander/IAddressBook/SearchSkypeDirectoryResponse" 
message="tns:SearchSkypeDirectorySoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="ProvideSkypeSearchFeedback">
      <wsdl:input wsaw:Action="DistributionListExpander/IAddressBook/ProvideSkypeSearchFeedback" 
message="tns:ProvideSkypeSearchFeedbackSoapIn"/>
      <wsdl:output 
wsaw:Action="DistributionListExpander/IAddressBook/ProvideSkypeSearchFeedbackResponse" 
message="tns:ProvideSkypeSearchFeedbackSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="AddressBookWebTicketBearer" type="tns:IAddressBook">
    <wsp:PolicyReference URI="#AddressBookWebTicketBearer_policy"/>
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="SearchAbEntry">
      <soap:operation soapAction="DistributionListExpander/IAddressBook/SearchAbEntry"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ExpandDistributionList">
      <soap:operation soapAction="DistributionListExpander/ExpandDistributionList"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SearchSkypeDirectory">
      <soap:operation soapAction="DistributionListExpander/IAddressBook/SearchSkypeDirectory"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ProvideSkypeSearchFeedback">
      <soap:operation
soapAction="DistributionListExpander/IAddressBook/ProvideSkypeSearchFeedback"
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="AddressBookWinNegotiate" type="tns:IAddressBook">
    <wsp:PolicyReference URI="#AddressBookWinNegotiate_policy"/>
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="SearchAbEntry">
      <soap:operation soapAction="DistributionListExpander/IAddressBook/SearchAbEntry"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ExpandDistributionList">
      <soap:operation soapAction="DistributionListExpander/ExpandDistributionList"
style="document"/>
```

```
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SearchSkypeDirectory">
      <soap:operation soapAction="DistributionListExpander/IAddressBook/SearchSkypeDirectory"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ProvideSkypeSearchFeedback">
      <soap:operation
soapAction="DistributionListExpander/IAddressBook/ProvideSkypeSearchFeedback"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

7 Appendix B: Product Behavior

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

- Microsoft Office Communications Server 2007
- Microsoft Office Communications Server 2007 R2
- Microsoft Office Communicator 2007
- Microsoft Office Communicator 2007 R2
- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Skype for Business (formerly Lync 2013)
- Skype for Business
- Skype for Business Server

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

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

<1> Section 3.1.1: The directory service (DS) that is used by Microsoft is Active Directory Domain Services (AD DS).

<2> Section 3.1.4: Office Communications Server 2007, Office Communications Server 2007 R2, Lync Server 2010, Lync Server 2013: ProvideSkypeSearchFeedback operation is not supported.

<3> Section 3.1.4: Office Communications Server 2007, Office Communications Server 2007 R2, Lync Server 2010, Lync Server 2013: SearchSkypeDirectory operation is not supported.

<4> Section 3.1.4.1.4.1: By default, the server returns a MemberCountLimitExceeded error when the number of members in a distribution list exceeds 100. The server can provide a way for administrators to configure this limit.

<5> Section 3.1.4.1.4.1: Lync 2010, Lync Server 2010: This response code is not supported.

<6> Section 3.1.4.1.4.1: By default, the address book server returns a SimultaneousRequestLimitExceeded error when the number of simultaneous requests from an external user exceeds 50 or 100 for an internal user. This limit is configurable on the server and can be overridden via the API exposed by the server.

<7> Section 3.1.4.2: Office Communicator 2007, Office Communications Server 2007, Office Communicator 2007 R2, Office Communications Server 2007 R2: The SearchAbEntry operation is not supported.

<8> Section 3.1.4.2.3.11: Office Communications Server 2007, Office Communications Server 2007 R2, Lync Server 2010, Lync Server 2013:SourceNetwork property is not available in the AbEntry object.

<9> Section 3.1.4.3: Office Communications Server 2007, Office Communications Server 2007 R2, Lync Server 2010, Lync Server 2013: SearchSkypeDirectory operation is not supported.

<10> Section 3.1.4.4: Office Communications Server 2007, Office Communications Server 2007 R2, Lync Server 2010, Lync Server 2013: ProvideSkypeSearchFeedback operation is not supported.

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
3.1.4 Message Processing Events and Sequencing Rules	Added two new Operations (SearchSkypeDirectory and ProvideSkypeSearchFeedback)	N	Content update.
3.1.4.3 SearchSkypeDirectory	Added section that describes protocol for SearchSkypeDirectory webservice API.	Y	New content added.
3.1.4.4 ProvideSkypeSearchFeedback	Added details of ProvideSkypeSearchFeedback API.	Y	New content added.
7 Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

9 Index