[MS-RGSWS]:
Response Group Service Web Service Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
03/31/2010	0.1	Major	Initial Availability
04/30/2010	0.2	Editorial	Revised and edited the technical content
06/07/2010	0.3	Editorial	Revised and edited the technical content
06/29/2010	0.4	Editorial	Changed language and formatting in the technical content.
07/23/2010	0.4	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.0	Major	Significantly changed the technical content.
11/15/2010	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Response Group Service, which enables administrators to create and configure one or more small response groups for the purpose of routing and queuing incoming phone calls to one or more designated agents. These response groups can be deployed in departmental or workgroup environments. The service also enables agents to make outbound calls on behalf of a response group.

Typical usage scenarios include a general external call handler, such as a service fronting the receptionists of a company, or a small customer service desk. The incoming calls treatment can include a welcome message, an out-of-business hour message, and an IVR.

The service exposes several methods to enable agents to interact with the system. For example, one method enables the agent to sign in and out of agent groups, enabling control if the service can send the agent calls.

This document specifies the procedure to access agent information.

1.1 Glossary

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

authentication
globally unique identifier (GUID)
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Secure Sockets Layer (SSL)
security identifier (SID)

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

agent
in-band provisioning
Session Initiation Protocol (SIP) address
SOAP (Simple Object Access Protocol)
SOAP body
SOAP envelope
SOAP message
SOAP operation
URL (Uniform Resource Locator)
WSDL (Web Services Description Language)
WSDL message
XML namespace
XML namespace prefix
XML Schema

The following terms are specific to this document:

response group: An object that is used to route and queue incoming calls to a collection of agents who were designated to handle calls from a Response Group Service.

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

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-OCAUTHWS] Microsoft Corporation, "OC Authentication Web Service Protocol Specification", March 2010.

[MS-SIPREGE] Microsoft Corporation, "Session Initiation Protocol (SIP) Registration Extensions", June 2008.

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

[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] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, http://www.w3.org/TR/REC-xml-names/

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "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-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary", June 2008.

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

1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to access **agent** information exposed by a protocol server. The protocol client issues requests to a protocol server. The protocol server receives, processes, and responds to the requests of protocol clients.

This protocol provides the following functionality:

- Indicates if a user is an agent.
- Retrieves agent information.

  The protocol client can get information about the current user from the protocol server through the Web service described in this protocol.
- Retrieves the list of agent groups.

  The protocol client can retrieve the list of agent groups the currently logged-in user is part of. The object obtained contains information about the group.
- Signs in and signs out of agent groups.

  The protocol client can send commands to the Web service described in this protocol to sign in and sign out the user from agent groups.
- Retrieves the list of active **response groups**.

  The protocol client can retrieve the list of response groups the user is part of through agent groups. The object obtained contains information about the response groups.

The Web service methods are documented in detail in section 6.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP (Simple Object Access Protocol)** message protocol for formatting requests and responses, as described in [SOAP1.2/1] and [SOAP1.2/2]. It transmits messages using the **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

The following figure shows the underlying messaging and transport stack that the protocol uses.

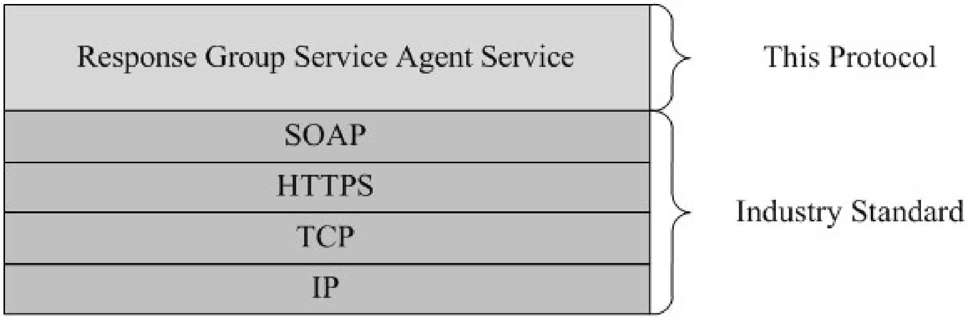

Figure 1: Underlying message and transport stack

1.5 Prerequisites/Preconditions

The protocol client can obtain the **Uniform Resource Locator (URL)** of the protocol server by using the mechanism described in [MS-SIPREGE] section 2.2.2.

The protocol requires that the protocol client has the correct permissions to call the methods on the protocol server.

The protocol client and server are described in [MS-OCAUTHWS].

1.6 Applicability Statement

This protocol is used to retrieve agent information.

This protocol requires authentication (2).

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTPS. The client can obtain the address of the protocol server via an **in-band provisioning** response, as specified in [MS-SIPREGE] section 2.2.2.5.1.

2.2 Common Message Syntax

This section contains common structures used by this protocol. The syntax of the structures uses **XML Schemas**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and the **WSDL (Web Services Description Language)**, as specified in [WSDL].

2.2.1 Namespaces

This protocol specifies and references **XML namespace** using the mechanisms specified in [XMLNS]. Although this document associates an **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

The following table specifies which XML namespace prefix is associated with each XML namespace that is used.

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

2.2.2 Messages

None.

2.2.3 Elements

None.

2.2.4 Complex Types

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

Complex type	Description
AcdAgent	Information about the authenticated user.

Complex type	Description
AcdGroup	Information about a group.
ArrayOfAcdAgent	A list of agents belonging to a group.
ArrayOfAcdGroup	A list of **AcdGroups**.
ArrayOfGuid	A list of **globally unique identifier (GUID)** instances.
ArrayOfResponseGroupEntry	A list of **ResponseGroupEntry**.
ResponseGroupEntry	Information about a response group.

2.2.4.1 AcdAgent

The **AcdAgent** complex type contains information related to an authenticated user.

```
<s:complexType name="AcdAgent">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Id" type="s1:guid" />
    <s:element minOccurs="0" maxOccurs="1" name="UserSid" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="SipAddress" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
  </s:sequence>
</s:complexType>
```

Id: Unique identifier identifying the user.

UserSid: The **security identifier (SID)** of the authenticated user.

SipAddress: **Session Initiation Protocol (SIP) address** of the authenticated user.

DisplayName: Display name of the authenticated user.

2.2.4.2 AcdGroup

The **AcdGroup** complex type contains information related to an agent group.

```
<s:complexType name="AcdGroup">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Id" type="s1:guid" />
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="NumberOfAgents" type="s:int" />
    <s:element minOccurs="1" maxOccurs="1" name="CanSignIn" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="SignInState" type="tns:SignInState" />
    <s:element minOccurs="0" maxOccurs="1" name="AllAgents"     type="tns:ArrayOfAcdAgent" />
    <s:element minOccurs="1" maxOccurs="1" name="NumberOfCallsWaiting" type="s:int" />
    <s:element minOccurs="1" maxOccurs="1" name="LongestWaitingTime" type="s:int" />

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

Id: Unique identifier identifying the group.

Name: Name of the group.

NumberOfAgents: Number of agents who are members of the group. **NumberOfAgents** SHOULD be greater or equal to zero.

CanSignIn: Boolean that indicates if the group supports sign in/sign out.

SignInState: State of the current user for this group.

AllAgents: List of agents who are members of the group.

NumberOfCallsWaiting: Number of calls waiting in the queues served by the group.

LongestWaitingTime: The longest waiting time in seconds among all the calls waiting in the queues served by the group.

2.2.4.3 ArrayOfAcdAgent

The **ArrayOfAcdAgent** complex type is a list of agents who are members of a group.

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

AcdAgent: Information related to the current authenticated user.

2.2.4.4 ArrayOfAcdGroup

The **ArrayOfAcdGroup** complex type is a list of **AcdGroup**s the current authenticated user is part of.

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

AcdGroup: See section 2.2.4.2.

2.2.4.5 ArrayOfGuid

The **ArrayOfGuid** complex type is a list of GUIDs representing groups.

```
<s:complexType name="ArrayOfGuid">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="guid" type="s1:guid" />
  </s:sequence>
</s:complexType>
```

guid: Identifier of a group.

2.2.4.6 ArrayOfResponseGroupEntry

The **ArrayOfResponseGroupEntry** complex group is a list of **ResponseGroupEntry** types the current authenticated user is part of.

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

ResponseGroupEntry: See section 2.2.4.7.

2.2.4.7 ResponseGroupEntry

The **ResponseGroupEntry** complex type contains information related to a response group.

```
<s:complexType name="ResponseGroupEntry">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Uri" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="IsAnonymized" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsOutboundAllowed" type="s:boolean" />
  </s:sequence>
</s:complexType>
```

Uri: Session Initiation Protocol (SIP) address of the response group.

DisplayName: Display name of the response group.

IsAnonymized: Boolean indicating if the response group supports anonymization.

IsOutboundAllowed: Boolean indicating if the response group allows its agent to make outbound calls.

2.2.5 Simple Types

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

Simple type	Description
SignInState	The sign-in state of an agent in a given group.

2.2.5.1 SignInState

The **SignInState** is a simple type representing the state of a user in a given group.

```
<s:simpleType name="SignInState">
  <s:restriction base="s:string">
    <s:enumeration value="SignedIn" />
    <s:enumeration value="SignedOut" />
```

```
      <s:enumeration value="Unknown" />
    </s:restriction>
  </s:simpleType>
```

SignedIn: The user is currently signed in to the group.

SignedOut: The user is currently signed out of the group.

Unknown: The user’s state cannot be retrieved.

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 group attribute definitions.

3 Protocol Details

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.

There is no order in which methods are required to be called, but the sign in methods **SignIn**, **SignInMultiple**, **SignOut**, and **SignOutMultiple** require as input valid GUIDs that can be retrieved using the **GetGroups** method. Also, if the user is not an agent, any method other than **IsAgent** SHOULD NOT return any data.

3.1 Server Details

This protocol does not require the protocol server to keep any states and information about the protocol client, unless required by the **authentication (2)** mechanism defined in [MS-OCAUTHWS] section 3.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table lists the **SOAP operations** specified in this protocol.

Operation	Description
GetActiveResponseGroups	Retrieve the list of response groups the authenticated user is part of.
GetAgent	Retrieve the information about the authenticated agent.
GetGroups	Retrieve the groups the authenticated user is part of.
IsAgent	Indicate if the authenticated user is an agent or not.
SignIn	Sign in the authenticated user to the specified group.
SignInMultiple	Sign in the authenticated user to the specified groups.
SignOut	Sign out the authenticated user from the specified group.
SignOutMultiple	Sign out the authenticated user from the specified groups.

3.1.4.1 GetActiveResponseGroups

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

```
<wsdl:operation name="GetActiveResponseGroups">
  <wsdl:input message="tns:GetAgentSoapIn" />
  <wsdl:output message="tns:GetAgentSoapOut" />
</wsdl:operation>
```

3.1.4.1.1 Messages

The **WSDL message** definitions in the following subsections are specific to the **GetActiveResponseGroups** operation.

3.1.4.1.1.1 GetActiveResponseGroupsSoapIn

The **GetActiveResponseGroupsSoapIn** **SOAP message** is a request that is sent from the protocol client to retrieve the list of response groups the user is part of. The request information MUST be captured in the **GetActiveResponseGroups** element in the **SOAP body** of the message. The **GetActiveResponseGroups** element is specified in section 3.1.4.1.2.1.

3.1.4.1.1.2 GetActiveResponseGroupsSoapOut

The **GetActiveResponseGroupsSoapOut** SOAP message is a response that is sent by the protocol server. This message contains information about the response groups the user is part of. The information is included in the complex type **ArrayOfResponseGroupEntry**, which is specified in section 2.2.4.6.

3.1.4.1.2 Complex Types

The following table lists the XML Schema complex type definitions specific to the **GetActiveResponseGroups** operation.

Complex type	Description
GetActiveResponseGroups	The overall container of the request to retrieve response groups.
GetActiveResponseGroupsResponse	The overall container of the response to the request to retrieve response groups.

3.1.4.1.2.1 GetActiveResponseGroups

The **GetActiveResponseGroups** complex type is the overall container of the information that is sent in the SOAP request to retrieve the response groups the user is part of. A protocol client MUST adhere to the following schema.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="GetActiveResponseGroups">
    <s:complexType />
  </s:element>
</s:schema>
```

3.1.4.1.2.2 GetActiveResponseGroupsResponse

The **GetActiveResponseGroupsResponse** complex type is the overall container in the response to the **GetActiveResponseGroups** request. **GetActiveResponseGroups** encapsulates the information about the authenticated user. A protocol client MUST adhere to the following schema for this complex type within the **SOAP envelope**.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="GetActiveResponseGroupsResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="0" maxOccurs="1" name=" GetActiveResponseGroupsResult"
type="tns:ArrayOfResponseGroupEntry" />
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.2 GetAgent

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

```
<wsdl:operation name="GetAgent">
      <wsdl:input message="tns:GetAgentSoapIn" />
      <wsdl:output message="tns:GetAgentSoapOut" />
</wsdl:operation>
```

3.1.4.2.1 Messages

The WSDL message definitions in the following subsections are specific to the **GetAgent** operation.

3.1.4.2.1.1 GetAgentSoapIn

The **GetAgentSoapIn** SOAP message is a request that is sent from the protocol client to retrieve information about the authenticated user. The request information MUST be captured in the **GetAgent** element in the SOAP body of the message. The **GetAgent** element is specified in section 3.1.4.2.2.1.

3.1.4.2.1.2 GetAgentSoapOut

The **GetAgentSoapOut** SOAP message is a response that is sent by the protocol server. This message contains information about the authenticated user if the user is an agent. The information is included in the complex type **AcdAgent**, which is specified in section 2.2.4.1. If the user is not an agent, the response SHOULD NOT contain any complex type.

3.1.4.2.2 Complex Types

The following table lists the XML Schema complex type definitions specific to the **GetAgent** operation.

Complex type	Description
GetAgent	The overall container of the request to retrieve agent information.
GetAgentResponse	The overall container of the response to the request to retrieve agent information.

3.1.4.2.2.1 GetAgent

The **GetAgent** complex type is the overall container of the information that is sent in the SOAP request to retrieve agent information. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="GetAgent">
    <s:complexType />
  </s:element>
</s:schema>
```

3.1.4.2.2.2 GetAgentResponse

The **GetAgentResponse** complex type is the overall container in the response to the **GetAgent** request. **GetAgentResponse** encapsulates the information about the authenticated user. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="GetAgentResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="0" maxOccurs="1" name="GetAgentResult" type="tns:AcdAgent" />
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.3 GetGroups

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

```
<wsdl:operation name="GetGroups">
      <wsdl:input message="tns:GetGroupsSoapIn" />
      <wsdl:output message="tns:GetGroupsSoapOut" />
</wsdl:operation>
```

3.1.4.3.1 Messages

The WSDL message definitions in the following subsections are specific to the **GetGroups** operation.

3.1.4.3.1.1 GetGroupsSoapIn

The **GetGroupsSoapIn** SOAP message is a request that is sent from the protocol client to retrieve the groups of an authenticated user. The request information MUST be captured in the **GetGroups** element in the SOAP body of the message. The **GetGroups** element is specified in section 3.1.4.3.2.1.

3.1.4.3.1.2 GetGroupsSoapOut

The **GetGroupsSoapOut** SOAP message is a response that is sent by the protocol server. This message contains a list of groups that the agent is member of. The information is included in the complex type **ArrayOfAcdGroup**, which is specified in section 2.2.4.4. If the agent is not part of any group, the **ArrayOfAcdGroup** complex type SHOULD be empty.

3.1.4.3.2 Complex Types

The following table lists XML Schema complex type definitions specific to the **GetGroups** operation.

Complex type	Description
GetGroups	The overall container of the request to retrieve agent groups.
GetGroupsResponse	The overall container of the response to the request to retrieve agent groups.

3.1.4.3.2.1 GetGroups

The **GetGroups** complex type is the overall container of the information that is sent in the SOAP request to retrieve agent groups. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="GetGroups">
    <s:complexType />
  </s:element>
</s:schema>
```

3.1.4.3.2.2 GetGroupsResponse

The **GetGroupsResponse** complex type is the overall container in the response to the **GetGroups** request. **GetGroupsResponse** encapsulates the information about the groups an agent is part of. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="GetGroupsResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="0" maxOccurs="1" name="GetGroupsResult" type="tns:ArrayOfAcdGroup"
/>
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.4 IsAgent

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

```
<wsdl:operation name="IsAgent">
      <wsdl:input message="tns:IsAgentSoapIn" />
      <wsdl:output message="tns:IsAgentSoapOut" />
</wsdl:operation>
```

3.1.4.4.1 Messages

The WSDL message definitions in the following subsections are specific to the **IsAgent** operation.

3.1.4.4.1.1 IsAgentSoapIn

The **IsAgentSoapIn** SOAP message is a request that is sent from the protocol client to retrieve information about whether the user is an agent. The request information MUST be captured in the **IsAgent** element in the SOAP body of the message. The **IsAgent** element is specified in section 3.1.4.4.2.1.

3.1.4.4.1.2 IsAgentSoapOut

The **IsAgentSoapOut** SOAP message is a response that is sent by the protocol server. This message MUST contain a Boolean that indicates if the user is an agent or not.

3.1.4.4.2 Complex Types

The following table lists XML Schema complex type definitions specific to the **IsAgent** operation.

Complex type	Description
IsAgent	The overall container of the request to retrieve agent state.
IsAgentResponse	The overall container of the response to the request to retrieve agent state.

3.1.4.4.2.1 IsAgent

The **IsAgent** complex type is the overall container of the information that is sent in the SOAP request to retrieve agent state. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="IsAgent">
    <s:complexType />
  </s:element>
</s:schema>
```

3.1.4.4.2.2 IsAgentResponse

The **IsAgentResponse** complex type is the overall container in the response to the **IsAgent** request. **IsAgentResponse** encapsulates the information about the agent state. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="IsAgentResponse">
```

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

3.1.4.5 SignIn

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

```
<wsdl:operation name="SignIn">
      <wsdl:input message="tns:SignInSoapIn" />
      <wsdl:output message="tns:SignInSoapOut" />
</wsdl:operation>
```

3.1.4.5.1 Messages

The WSDL message definitions in the following subsections are specific to the **SignIn** operation.

3.1.4.5.1.1 SignInSoapIn

The **SignInSoapIn** SOAP message is a request that is sent from the protocol client to sign in the user in the specified group. The request information MUST be captured in the **SignIn** element in the SOAP body of the message. The **SignIn** element is specified in section 3.1.4.5.2.1.

3.1.4.5.1.2 SignInSoapOut

This SOAP message is a response that is sent by the protocol server. This message MUST contain a Boolean indicating if the operation has succeeded.

3.1.4.5.2 Complex Types

The following table lists XML Schema complex type definitions specific to the **SignIn** operation.

Complex type	Description
SignIn	The overall container of the request to sign in the user to a group.
SignInResponse	The overall container of the response to the request to sign in the user to a group.

3.1.4.5.2.1 SignIn

The **SignIn** complex type is the overall container of the information that is sent in the SOAP request to sign in the user in a group. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignIn">
    <s:complexType>
```

```
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="groupId" type="s1:guid" />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.5.2.2 SignInResponse

The **SignInResponse** complex type is the overall container in the response to the **SignIn** request. **SignInResponse** contains the result of the operation. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignInResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="1" maxOccurs="1" name="SignInResult" type="s:boolean" />
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.6 SignInMultiple

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

```
<wsdl:operation name="SignInMultiple">
      <wsdl:input message="tns:SignInMultipleSoapIn" />
      <wsdl:output message="tns:SignInMultipleSoapOut" />
</wsdl:operation>
```

3.1.4.6.1 Messages

The WSDL message definitions in the following subsections are specific to the **SignInMultiple** operation.

3.1.4.6.1.1 SignInMultipleSoapIn

The **SignInMultipleSoapIn** SOAP message is a request that is sent from the protocol client to sign in the user in the specified groups. The request information MUST be captured in the **SignInMultiple** element in the SOAP body of the message. The **SignInMultiple** element is specified in section 3.1.4.6.2.1.

3.1.4.6.1.2 SignInMultipleSoapOut

The **SignInMultipleSoapOut** SOAP message is a response that is sent by the protocol server. This message MUST contain a Boolean indicating if the operation succeeded.

3.1.4.6.2 Complex Types

The following table lists XML Schema complex type definitions specific to the **SignInMultiple** operation.

Complex type	Description
SignInMultiple	The overall container of the request to sign in the user in multiple groups.
SignInMultipleResponse	The overall container of the response to the request to sign in the user in multiple groups.

3.1.4.6.2.1 SignInMultiple

The **SignInMultiple** complex type is the overall container of the information that is sent in the SOAP request to sign in the user in multiple groups. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified" 
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignInMultiple">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="groupIds" type="tns:ArrayOfGuid" />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.6.2.2 SignInMultipleResponse

The **SignInMultipleResponse** complex type is the overall container in the response to the **SignInMultiple** request. **SignInMultipleResponse** contains the result of the operation. If the operation partially succeeded, the return value SHOULD be set to "false". Information about the current sign-in state can be retrieved using the **GetGroups** message, as defined in section 3.1.4.3. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified" 
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignInMultipleResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="1" maxOccurs="1" name="SignInMultipleResult" type="s:boolean" />
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.7 SignOut

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

```
<wsdl:operation name="SignOut">
      <wsdl:input message="tns:SignOutSoapIn" />
      <wsdl:output message="tns:SignOutSoapOut" />
</wsdl:operation>
```

3.1.4.7.1 Messages

The following WSDL message definitions are specific to the **SignOut** operation.

3.1.4.7.1.1 SignOutSoapIn

The **SignOutSoapIn** SOAP message is a request that is sent from the protocol client to sign out the user from the specified group. The request information MUST be captured in the **SignOut** element in the SOAP body of the message. The **SignOut** element is specified in section 3.1.4.7.2.1.

3.1.4.7.1.2 SignOutSoapOut

The **SignOutSoapOut** SOAP message is a response that is sent by the protocol server. This message MUST contain a Boolean indicating if the operation succeeded.

3.1.4.7.2 Complex Types

The following table lists XML Schema complex type definitions specific to the **SignOut** operation.

Complex type	Description
SignOut	The overall container of the request to sign out the user from a group.
SignOutResponse	The overall container of the response to the request to sign out the user from a group.

3.1.4.7.2.1 SignOut

The **SignOut** complex type is the overall container of the information that is sent in the SOAP request to sign out the user from a group. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignOut">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="groupId" type="s1:guid" />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.7.2.2 SignOutResponse

The **SignOutResponse** complex type is the overall container in the response to the **SignOut** request. **SignOutResponse** contains the result of the operation. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignOutResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="1" maxOccurs="1" name="SignOutResult" type="s:boolean" />
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.8 SignOutMultiple

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

```
<wsdl:operation name="SignOutMultiple">
      <wsdl:input message="tns:SignOutMultipleSoapIn" />
      <wsdl:output message="tns:SignOutMultipleSoapOut" />
</wsdl:operation>
```

3.1.4.8.1 Messages

The WSDL message definitions in the following subsections are specific to the **SignOutMultiple** operation.

3.1.4.8.1.1 SignOutMultipleSoapIn

The **SignOutMultipleSoapIn** SOAP message is a request that is sent from the protocol client to sign out the user from the specified groups. The request information MUST be captured in the **SignOutMultiple** element in the SOAP body of the message. The **SignOutMultiple** element is specified in section 3.1.4.8.2.1.

3.1.4.8.1.2 SignOutMultipleSoapOut

The **SignOutMultipleSoapOut** SOAP message is a response that is sent by the protocol server. This message MUST contain a Boolean indicating if the operation succeeded.

3.1.4.8.2 Complex Types

The following table lists XML Schema complex type definitions specific to the **SignOutMultiple** operation.

Complex type	Description
SignOutMultiple	The overall container of the request to sign out the user from multiple groups.
SignOutMultipleResponse	The overall container of the response to the request to sign out the user from multiple groups.

3.1.4.8.2.1 SignOutMultiple

The **SignOutMultiple** complex type is the overall container of the information that is sent in the SOAP request to sign out the user from multiple groups. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignOutMultiple">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="groupIds" type="tns:ArrayOfGuid" />
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.4.8.2.2 SignOutMultipleResponse

The **SignOutMultipleResponse** complex type is the overall container in the response to the **SignOutMultiple** request. **SignOutMultipleResponse** contains the result of the operation. If the operation partially succeeded, the return value SHOULD be set to "false". Information about the current sign-in state can be retrieved using the **GetGroups** message, as defined in section 3.1.4.3. A protocol client MUST adhere to the following schema for this complex type within the SOAP envelope.

```
<s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
  <s:element name="SignOutMultipleResponse">
    <s:complexType>
   <s:sequence>
     <s:element minOccurs="1" maxOccurs="1" name=" SignOutMultipleResult" type="s:boolean" />
   </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details

None.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

4.1 Example of a Successful GetActiveResponseGroups Request and Response

The following example is a **GetActiveResponseGroups** request. This request is sent from a client to the server as a SOAP HTTPS request.

```
<soap:Body>
  <GetActiveResponseGroups
xmlns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy"/>
</soap:Body>
```

This request results in the following successful SOAP HTTPS response.

```
<soap:Body>
  <GetActiveResponseGroupsResponse
xmlns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
    <GetActiveResponseGroupsResult>
      <ResponseGroupEntry>
        <Uri>sip:helpdesk@contoso.com</Uri>
        <DisplayName>Contoso Helpdesk</DisplayName>
        <IsAnonymized>true</IsAnonymized>
        <IsOutboundAllowed>true</IsOutboundAllowed>
      </ResponseGroupEntry>
      <ResponseGroupEntry>
        <Uri>sip:hr@contoso.com</Uri>
        <DisplayName>Contoso HR</DisplayName>
        <IsAnonymized>true</IsAnonymized>
        <IsOutboundAllowed>false</IsOutboundAllowed>
      </ResponseGroupEntry>
    </GetActiveResponseGroupsResult>
  </GetActiveResponseGroupsResponse>
</soap:Body>
```

4.2 Example of a Successful SignIn Request and Response

The following example is a **SignIn** request. This request is sent from a client to the server as a SOAP HTTPS request.

```
<soap:Body>
  <SignIn xmlns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
    <groupId>0762e30f-a76c-4c75-9390-4c07a9f4e51f</groupId>
 </SignIn>
</soap:Body>
```

This request results in the following successful SOAP HTTPS response.

```
<soap:Body>
  <SignInResponse
xmlns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
    <SignInResult>true</SignInResult>
  </SignInResponse>
</soap:Body>
```

4.3 Example of an Unsuccessful SignOut Request and Response

The following example is a **SignOut** request. This request is sent from a client to the server as a SOAP HTTPS request.

```
<soap:Body>
  <SignOut xmlns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
    <groupId>0762e30f-a76c-4c75-9390-4c07a9f4e51f</groupId>
  </SignOut>
</soap:Body>
```

This request results in the following SOAP HTTPS response.

```
<soap:Body>
  <SignOutResponse
xmlns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
    <SignOutResult>false</SignOutResult>
  </SignOutResponse>
</soap:Body>
```

5 Security

5.1 Security Considerations for Implementers

This protocol allows **Hypertext Transfer Protocol (HTTP)** connections only over **Secure Sockets Layer (SSL)**. Users are authenticated using the mechanism described in [MS-OCAUTHWS].

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided below:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy"
xmlns:s1="http://microsoft.com/wsdl/types/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
  <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy">
    <s:import namespace="http://microsoft.com/wsdl/types/" />
    <s:element name="IsAgent">
      <s:complexType />
    </s:element>
    <s:element name="IsAgentResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="IsAgentResult" type="s:boolean" />
        </s:sequence>
       </s:complexType>
      </s:element>
    <s:element name="GetAgent">
      <s:complexType />
    </s:element>
    <s:element name="GetAgentResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetAgentResult" type="tns:AcdAgent" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="AcdAgent">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="Id" type="s1:guid" />
        <s:element minOccurs="0" maxOccurs="1" name="UserSid" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="SipAddress" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
      </s:sequence>
    </s:complexType>
    <s:element name="GetGroups">
      <s:complexType />
    </s:element>
    <s:element name="GetGroupsResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetGroupsResult"
type="tns:ArrayOfAcdGroup" />
         </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOfAcdGroup">
```

```
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="AcdGroup" nillable="true"
type="tns:AcdGroup" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="AcdGroup">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="Id" type="s1:guid" />
        <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="NumberOfAgents" type="s:int" />
        <s:element minOccurs="1" maxOccurs="1" name="CanSignIn" type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1" name="SignInState" type="tns:SignInState" />
        <s:element minOccurs="0" maxOccurs="1" name="AllAgents" type="tns:ArrayOfAcdAgent" />
        <s:element minOccurs="1" maxOccurs="1" name="NumberOfCallsWaiting" type="s:int" />
        <s:element minOccurs="1" maxOccurs="1" name="LongestWaitingTime" type="s:int" />

      </s:sequence>
    </s:complexType>
    <s:simpleType name="SignInState">
      <s:restriction base="s:string">
        <s:enumeration value="SignedIn" />
        <s:enumeration value="SignedOut" />
        <s:enumeration value="Unknown" />
      </s:restriction>
    </s:simpleType>
    <s:complexType name="ArrayOfAcdAgent">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="AcdAgent" nillable="true"
type="tns:AcdAgent" />
      </s:sequence>
    </s:complexType>
    <s:element name="SignIn">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="groupId" type="s1:guid" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SignInResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SignInResult" type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SignOut">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="groupId" type="s1:guid" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SignOutResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SignOutResult" type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SignInMultiple">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="groupIds" type="tns:ArrayOfGuid" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOfGuid">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="guid" type="s1:guid" />
      </s:sequence>
    </s:complexType>
    <s:element name="SignInMultipleResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SignInMultipleResult" type="s:boolean"
/>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SignOutMultiple">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="groupIds" ype="tns:ArrayOfGuid" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SignOutMultipleResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SignOutMultipleResult"
type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetActiveResponseGroups">
      <s:complexType />
    </s:element>
    <s:element name="GetActiveResponseGroupsResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetActiveResponseGroupsResult"
type="tns:ArrayOfResponseGroupEntry" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOfResponseGroupEntry">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="ResponseGroupEntry"
nillable="true" type="tns:ResponseGroupEntry" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ResponseGroupEntry">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Uri" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="IsAnonymized" type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1" name="IsOutboundAllowed" type="s:boolean" />
      </s:sequence>
    </s:complexType>
  </s:schema>
  <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/">
    <s:simpleType name="guid">
      <s:restriction base="s:string">
        <s: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}" />
      </s:restriction>
    </s:simpleType>
  </s:schema>
</wsdl:types>
<wsdl:message name="IsAgentSoapIn">
  <wsdl:part name="parameters" element="tns:IsAgent" />
</wsdl:message>
<wsdl:message name="IsAgentSoapOut">
  <wsdl:part name="parameters" element="tns:IsAgentResponse" />
</wsdl:message>
<wsdl:message name="GetAgentSoapIn">
  <wsdl:part name="parameters" element="tns:GetAgent" />
</wsdl:message>
<wsdl:message name="GetAgentSoapOut">
  <wsdl:part name="parameters" element="tns:GetAgentResponse" />
</wsdl:message>
<wsdl:message name="GetGroupsSoapIn">
  <wsdl:part name="parameters" element="tns:GetGroups" />
</wsdl:message>
<wsdl:message name="GetGroupsSoapOut">
  <wsdl:part name="parameters" element="tns:GetGroupsResponse" />
</wsdl:message>
<wsdl:message name="SignInSoapIn">
  <wsdl:part name="parameters" element="tns:SignIn" />
</wsdl:message>
<wsdl:message name="SignInSoapOut">
  <wsdl:part name="parameters" element="tns:SignInResponse" />
</wsdl:message>
<wsdl:message name="SignOutSoapIn">
  <wsdl:part name="parameters" element="tns:SignOut" />
</wsdl:message>
<wsdl:message name="SignOutSoapOut">
  <wsdl:part name="parameters" element="tns:SignOutResponse" />
</wsdl:message>
<wsdl:message name="SignInMultipleSoapIn">
  <wsdl:part name="parameters" element="tns:SignInMultiple" />
</wsdl:message>
<wsdl:message name="SignInMultipleSoapOut">
  <wsdl:part name="parameters" element="tns:SignInMultipleResponse" />
</wsdl:message>
<wsdl:message name="SignOutMultipleSoapIn">
  <wsdl:part name="parameters" element="tns:SignOutMultiple" />
</wsdl:message>
<wsdl:message name="SignOutMultipleSoapOut">
  <wsdl:part name="parameters" element="tns:SignOutMultipleResponse" />
</wsdl:message>
<wsdl:message name="GetActiveResponseGroupsSoapIn">
  <wsdl:part name="parameters" element="tns:GetActiveResponseGroups" />
</wsdl:message>
<wsdl:message name="GetActiveResponseGroupsSoapOut">
  <wsdl:part name="parameters" element="tns:GetActiveResponseGroupsResponse" />
```

```
</wsdl:message>
<wsdl:portType name="ProxyServiceSoap">
  <wsdl:operation name="IsAgent">
    <wsdl:input message="tns:IsAgentSoapIn" />
    <wsdl:output message="tns:IsAgentSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetAgent">
    <wsdl:input message="tns:GetAgentSoapIn" />
    <wsdl:output message="tns:GetAgentSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetGroups">
    <wsdl:input message="tns:GetGroupsSoapIn" />
    <wsdl:output message="tns:GetGroupsSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="SignIn">
    <wsdl:input message="tns:SignInSoapIn" />
    <wsdl:output message="tns:SignInSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="SignOut">
    <wsdl:input message="tns:SignOutSoapIn" />
    <wsdl:output message="tns:SignOutSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="SignInMultiple">
    <wsdl:input message="tns:SignInMultipleSoapIn" />
    <wsdl:output message="tns:SignInMultipleSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="SignOutMultiple">
    <wsdl:input message="tns:SignOutMultipleSoapIn" />
    <wsdl:output message="tns:SignOutMultipleSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetActiveResponseGroups">
    <wsdl:input message="tns:GetActiveResponseGroupsSoapIn" />
    <wsdl:output message="tns:GetActiveResponseGroupsSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ProxyServiceSoap" type="tns:ProxyServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="IsAgent">
    <soap:operation 
soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/IsAgent" 
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetAgent">
    <soap:operation 
soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/GetAgent
" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
```

```
    <wsdl:operation name="GetGroups">
      <soap:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/GetGroup
  s" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SignIn">
      <soap:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignIn"
  style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SignOut">
      <soap:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignOut"
  style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SignInMultiple">
      <soap:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignInMu
  ltiple" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SignOutMultiple">
      <soap:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignOutM
  ultiple" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetActiveResponseGroups">
      <soap:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/GetActiv
  eResponseGroups" 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="ProxyServiceSoap12" type="tns:ProxyServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="IsAgent">
      <soap12:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/IsAgent"
  style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAgent">
      <soap12:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/GetAgent
  " style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetGroups">
      <soap12:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/GetGroup
  s" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SignIn">
      <soap12:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignIn"
  style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SignOut">
      <soap12:operation
  soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignOut"
  style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SignInMultiple">
    <soap12:operation
soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignInMu
ltiple" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SignOutMultiple">
    <soap12:operation
soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/SignOutM
ultiple" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name=" GetActiveResponseGroups">
    <soap12:operation
soapAction="http://schemas.microsoft.com/acd/2007/12/Microsoft.Rtc.Acd.Clients.Proxy/GetActiv
eResponseGroups" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

7 Appendix B: Product Behavior

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

- Microsoft® Lync™ Server 2010
- Microsoft® Lync™ 2010

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

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

8 Change Tracking

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

9 Index