[MS-RSWSSFA]:

Report Server Web Service for SharePoint Forms Authentication: ReportServiceAuthentication

Intellectual Property Rights Notice for Open Specifications Documentation

Revision Summary

Date	Revision History	Revision Class	Comments
8/7/2009	0.1	Major	First release.
11/6/2009	0.1.1	Editorial	Changed language and formatting in the technical content.
3/5/2010	0.2	Minor	Clarified the meaning of the technical content.
4/21/2010	0.2.1	Editorial	Changed language and formatting in the technical content.
6/4/2010	0.3	Minor	Clarified the meaning of the technical content.
9/3/2010	0.4	Minor	Clarified the meaning of the technical content.
2/9/2011	0.4	None	No changes to the meaning, language, or formatting of the technical content.
7/7/2011	0.4	None	No changes to the meaning, language, or formatting of the technical content.
11/3/2011	0.4	None	No changes to the meaning, language, or formatting of the technical content.
1/19/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
2/23/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
3/27/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
5/24/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
6/29/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	0.4	None	No changes to the meaning, language, or formatting of the technical content.
3/26/2013	0.4	None	No changes to the meaning, language, or formatting of the technical content.
6/11/2013	0.4	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	0.4	None	No changes to the meaning, language, or formatting of the technical content.
12/5/2013	0.4	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2014	1.0	Major	Updated and revised the technical content.

Date	Revision History	Revision Class	Comments
5/20/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
5/10/2016	2.0	Major	Significantly changed the technical content.

Table of Contents

1   Introduction

The Report Server Web Service for SharePoint Forms Authentication: ReportServiceAuthentication protocol is an API protocol that is provided by the ReportServiceAuthentication web service for authenticating users against a **report server** when the SharePoint web application is configured for Forms Authentication.

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

1.1   Glossary

This document uses the following terms:

report server: A location on the network to which clients can connect by using SOAP over HTTP or SOAP over HTTPS to publish, manage, and execute reports.

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [SOAP1.2-1/2007] section 5.2 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.

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

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

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

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

1.2   References

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

1.2.1   Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

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

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

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation 27, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[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/

1.2.1.1 Prescriptive API References

[MSDN-SSRS] Microsoft Corporation, "Reporting Services (SSRS)", http://msdn.microsoft.com/en-us/library/ms159106.aspx

[MSDN-SQL2008RSWS] Microsoft Corporation,"ReportService2006 Namespace", http://msdn.microsoft.com/en-us/library/reportservice2006.aspx

1.2.2 Informative References

[MSDN-RSAM] Microsoft Corporation, "ReportServiceAuthentication Methods", http://msdn.microsoft.com/en-us/library/reportserviceauthentication.reportserviceauthentication_methods.aspx

[MSDN-RSAN] Microsoft Corporation, "ReportServiceAuthentication Namespace", http://msdn.microsoft.com/en-us/library/reportserviceauthentication.aspx

[MSDN-SoapHeader] Microsoft Corporation, "SoapHeader Class", http://msdn.microsoft.com/en-us/library/system.web.services.protocols.soapheader.aspx

[MSDN-SORSSIM] Microsoft Corporation, "Security Overview for Reporting Services in SharePoint Integrated Mode", http://msdn.microsoft.com/en-us/library/bb283324.aspx

[MSDN-SSRS] Microsoft Corporation, "Reporting Services (SSRS)", http://msdn.microsoft.com/en-us/library/ms159106.aspx

1.3 Overview

The ReportServiceAuthentication web service protocol [MSDN-RSAN] that is described in this document provides methods for retrieving the authentication mode and for authenticating users against a report server [MSDN-SSRS] when the SharePoint web application is configured for Forms Authentication [MSDN-SORSSIM].

Typically, the client establishes a connection with the server. After the connection is established by using the HTTP [RFC2616] or HTTPS [RFC2818] protocol, **SOAP messages** [SOAP1.1] [SOAP1.2-1/2007] are used to communicate between the client and the server.

The ReportServiceAuthentication web service protocol uses the security facilities that are built into HTTP or HTTPS for authentication and identification and for channel encryption negotiation. The protocol uses the facilities that are built into SOAP for specification of requests from client to server and for returning data from the server to the client. The following diagram depicts a (simplified) typical flow of communication in the protocol.
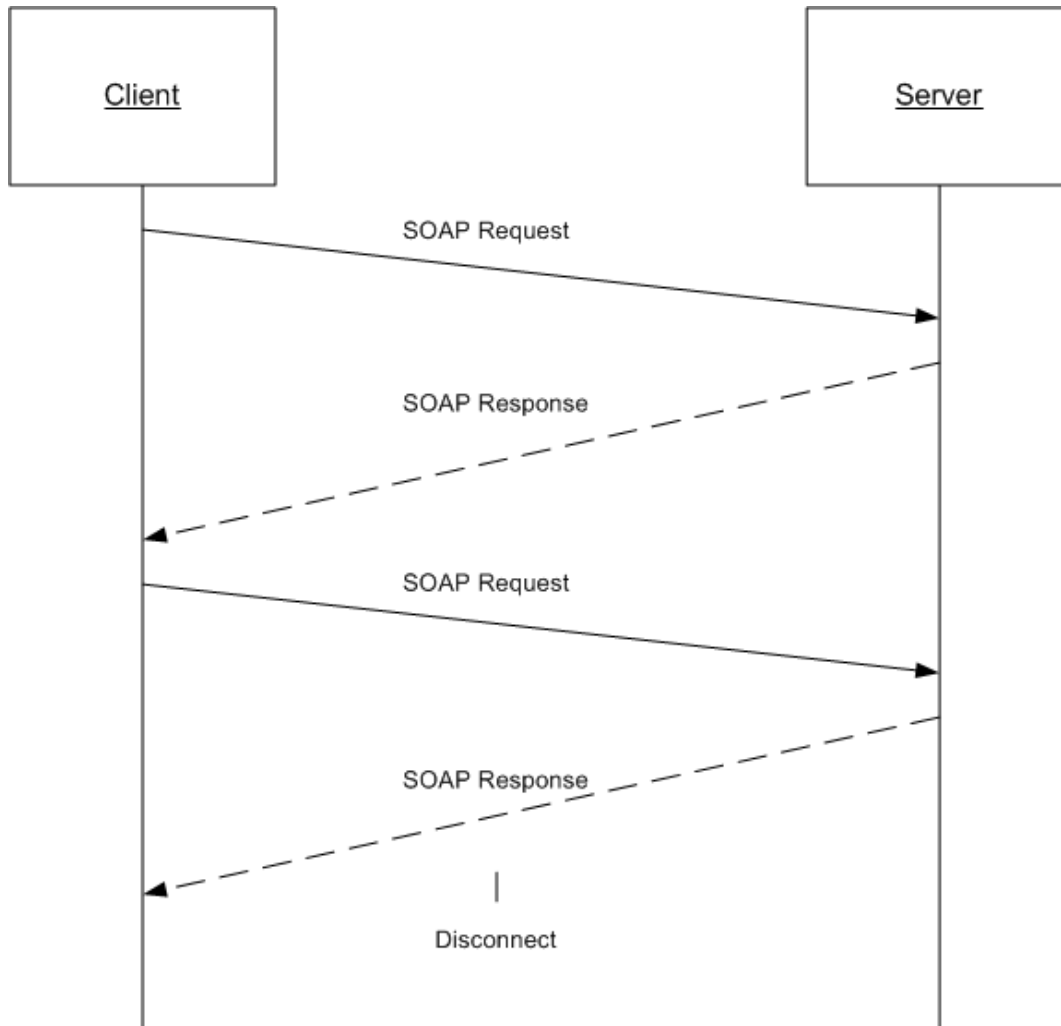
Figure 1: Communication flow for ReportServiceAuthentication

Each method in the protocol is a SOAP operation that accepts a set of parameters as a SOAP request and returns a set of values as a SOAP response. The client sends a request to the server through a SOAP request message, and the server sends return values to the client through a SOAP response message.

The following procedure is a simple example client/server exchange for authenticating a user.

1. The client calls the **LogonUser** method and passes in the user name and the password as method parameters.

2. The server authenticates the user. If authentication is successful, the server outputs the authentication cookie and its name and then returns true. If authentication is unsuccessful, the server returns false.

3. The client makes subsequent web requests, presenting the authentication cookie (through a different web protocol). The requests can then be authenticated by the server.

1.4 Relationship to Other Protocols

The ReportServiceAuthentication web service protocol uses SOAP over HTTP as shown in the following layering diagram.
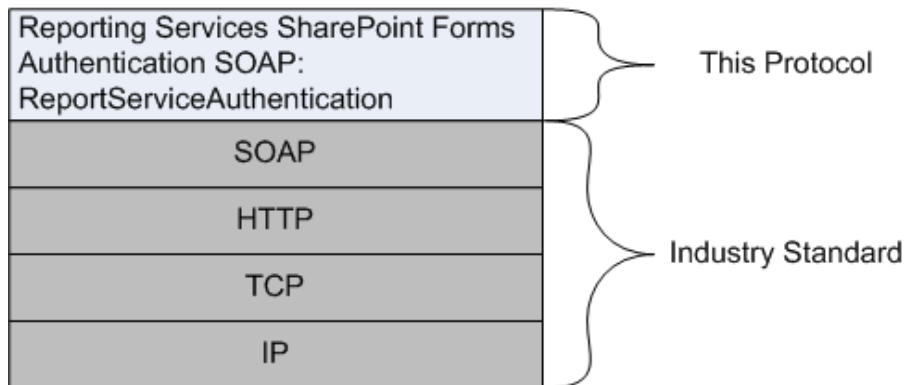
Figure 2: SOAP over HTTP

The ReportServiceAuthentication web service protocol uses SOAP over HTTPS as shown in the following layering diagram.
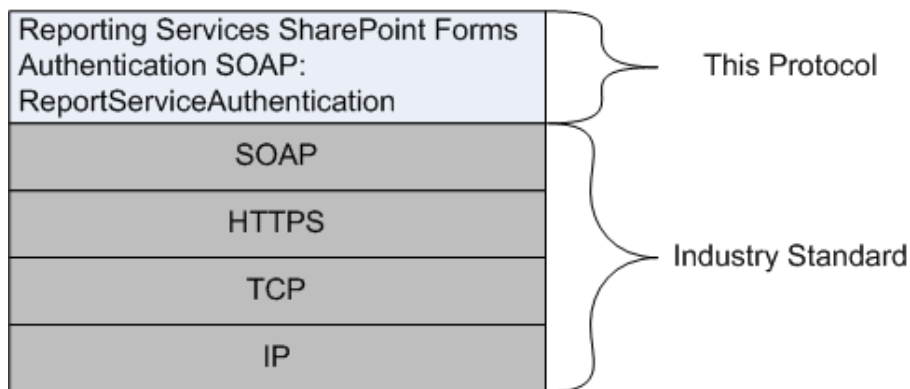
Figure 3: SOAP over HTTPS

1.5 Prerequisites/Preconditions

The ReportServiceAuthentication web service operates against a URL. The URL is identified by protocol clients. The protocol server endpoint is formed by appending "/ReportServiceAuthentication.asmx" to the URL, as in the following example:

http://www.contoso.com/ReportSite/_vti_bin/ReportServer/ReportServiceAuthentication.asmx

It is assumed that authentication has been performed by the underlying protocol. The account that accesses this web service is assigned with proper permissions to the site and its items.

1.6 Applicability Statement

The ReportServiceAuthentication web service protocol enables client applications to run and navigate reports on a report server.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

Supported Transports: The ReportServiceAuthentication web service protocol uses multiple transports with SOAP as specified in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTP and MUST support SOAP over HTTPS for securing communication with clients.

Protocol messages MUST be formatted as specified in section 4 of [SOAP1.1] or section 3 of [SOAP1.2-1/2007].<1>

2.2 Messages

This section defines messages that are used by this protocol. The syntax of the definitions uses XML Schema as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and it uses Web Services Description Language (WSDL) as defined in [WSDL].

The following information describes the steps by which the API is mapped to structures and operations in the web service.

Each class in the ReportServiceAuthentication web service [MSDN-RSAN], except the class that is derived from **System.Web.Services.Protocols.SoapHttpClientProtocol**, becomes one WSDL type. The following rules apply:

- Non-inherited public properties become elements in the type, with the following exceptions and additional rules:

    - Any property named *AnyAttr* becomes the XSD **anyAttribute** in the type rather than an element.

    - If the name of a property is the concatenation of the name of another property in the same class and "Specified", no element is created for the former property in the type. The **minOccurs** attribute of the element for the latter property is 0. For other properties, the **minOccurs** attribute is 1 unless the API data type is **String**, **Byte[]**, or an array type, in which case the **minOccurs** attribute is 0.

    - The **maxOccurs** attribute of the element is 1 for nonarray types and "unbounded" for array types.

- If the class inherits another class in the web service, the WSDL type of the inherited class becomes an XSD extension in the WSDL type corresponding to the inheriting class.

Each enumeration in the web service becomes one WSDL type as an XSD enumeration.

Each class that inherits **System.Web.Services.Protocols.SoapHeader** [MSDN-SoapHeader] becomes one **SOAP header**.

Each method in the class that is derived from **System.Web.Services.Protocols.SoapHttpClientProtocol** that is not derived from a base method in a base class or an interface becomes one **WSDL operation** with a corresponding SOAP operation, a request **WSDL message**, a response WSDL message, and corresponding WSDL types. The following rules apply:

- Out parameters and return values of the method become elements in the WSDL type for the response WSDL message.

- Other parameters of the method become elements in the WSDL type for the request WSDL message.

- If the method uses a SOAP header, the WSDL operation contains a WSDL message that has a single part of the WSDL type corresponding to the SOAP header.

An array of a class becomes a WSDL type with an element of the WSDL type corresponding to the class.

The primitive data types that are used by the API are mapped to WSDL XML Schema types as described in the following table.

API primitive data type	WSDL XML Schema type
Byte[]	base64Binary
Boolean	boolean
SByte	byte
Double	double
DateTime	dateTime
Decimal	decimal
Single	float
Int32	int
Int64	long
XmlQualifiedName	QName
Int16	short
String	string
Byte	unsignedByte
UInt32	unsignedInt
UInt64	unsignedLong
UInt16	unsignedShort

The result of this mapping can be found in the full WSDL in Appendix A of this specification.

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms that are 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/	[WSDL]
tns	http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices	Defined in Appendix A of this document
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

Prefix	Namespace URI	Reference
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[WSDL]
wsdl	http://schemas.xmlsoap.org/wsdl	[WSDL]

2.2.2 Methods

The following table summarizes the set of method definitions that are defined by this specification.

Reporting Services SharePoint Forms Authentication: ReportServiceAuthentication methods are defined in detail in [MSDN-RSAM].

Method	Description
GetAuthenticationMode	Gets the authentication scheme that is used by the SharePoint web application.
Logoff	Logs off the current user who is making web service requests.
LogonUser	Logs on a user and authenticates a user request to the Report Server web service.

2.2.3 Types

The following table summarizes the set of type definitions that are defined by this specification.

Reporting Services SharePoint Forms Authentication: ReportServiceAuthentication types are defined in detail in [MSDN-RSAN].

Type	Description
AuthenticationMode	Represents the different authentication schemes that are available for a SharePoint web application that is configured for Forms Authentication.

2.2.4 SOAP Headers

None.

3 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided here.

```
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservice
s" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">The Reporting Services
Web Service enables you to manage a report server and its contents including server settings,
security, reports, subscriptions, and data sources.</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservice
s">
      <s:element name="LogonUser">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="userName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="authority" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="LogonUserResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="LogonUserResult" type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="cookieName" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="Logoff">
        <s:complexType />
      </s:element>
      <s:element name="LogoffResponse">
        <s:complexType />
      </s:element>
      <s:element name="GetAuthenticationMode">
        <s:complexType />
      </s:element>
      <s:element name="GetAuthenticationModeResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetAuthenticationModeResult"
                                                  type="tns:AuthenticationMode" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:simpleType name="AuthenticationMode">
        <s:restriction base="s:string">
          <s:enumeration value="None" />
          <s:enumeration value="Windows" />
          <s:enumeration value="Passport" />
          <s:enumeration value="Forms" />
        </s:restriction>
      </s:simpleType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="LogonUserSoapIn">
    <wsdl:part name="parameters" element="tns:LogonUser" />
  </wsdl:message>
  <wsdl:message name="LogonUserSoapOut">
    <wsdl:part name="parameters" element="tns:LogonUserResponse" />
  </wsdl:message>
```

```
<wsdl:message name="LogoffSoapIn">
  <wsdl:part name="parameters" element="tns:Logoff" />
</wsdl:message>
<wsdl:message name="LogoffSoapOut">
  <wsdl:part name="parameters" element="tns:LogoffResponse" />
</wsdl:message>
<wsdl:message name="GetAuthenticationModeSoapIn">
  <wsdl:part name="parameters" element="tns:GetAuthenticationMode" />
</wsdl:message>
<wsdl:message name="GetAuthenticationModeSoapOut">
  <wsdl:part name="parameters" element="tns:GetAuthenticationModeResponse" />
</wsdl:message>
<wsdl:portType name="ReportServiceAuthenticationSoap">
  <wsdl:operation name="LogonUser">
    <wsdl:input message="tns:LogonUserSoapIn" />
    <wsdl:output message="tns:LogonUserSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="Logoff">
    <wsdl:input message="tns:LogoffSoapIn" />
    <wsdl:output message="tns:LogoffSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationMode">
    <wsdl:input message="tns:GetAuthenticationModeSoapIn" />
    <wsdl:output message="tns:GetAuthenticationModeSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ReportServiceAuthenticationSoap"
type="tns:ReportServiceAuthenticationSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="LogonUser">
    <soap:operation
soapAction="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices/Log
onUser" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="Logoff">
    <soap:operation
soapAction="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices/Log
off" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationMode">
    <soap:operation
soapAction="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices/Get
AuthenticationMode" 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="ReportServiceAuthenticationSoap12"
type="tns:ReportServiceAuthenticationSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="LogonUser">
```

```
      <soap12:operation
soapAction="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices/Log
onUser" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Logoff">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices/Log
off" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAuthenticationMode">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sqlserver/2006/03/15/reporting/reportingservices/Get
AuthenticationMode" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ReportServiceAuthentication">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">The Reporting Services
Web Service enables you to manage a report server and its contents including server settings,
security, reports, subscriptions, and data sources.</wsdl:documentation>
    <wsdl:port name="ReportServiceAuthenticationSoap"
binding="tns:ReportServiceAuthenticationSoap">
      <soap:address
location="http://sqlserver/sites/rosetta/_vti_bin/ReportServer/ReportServiceAuthentication.as
mx" />
    </wsdl:port>
    <wsdl:port name="ReportServiceAuthenticationSoap12"
binding="tns:ReportServiceAuthenticationSoap12">
      <soap12:address
location="http://sqlserver/sites/rosetta/_vti_bin/ReportServer/ReportServiceAuthentication.as
mx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

4   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 SQL Server 2005 Service Pack 2 (SP2)

- Microsoft SQL Server 2008

- Microsoft SQL Server 2008 R2

- Microsoft SQL Server 2012

- Microsoft SQL Server 2014

- Microsoft SQL Server 2016

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

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

<1> Section 2.1: Except in SQL Server 2005 SP2 Reporting Services, Microsoft SQL Server Reporting Services supports both SOAP 1.1 [SOAP1.1] and SOAP 1.2 [SOAP1.2-1/2007].

5   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
4 Appendix B: Product Behavior	Added SQL Server 2016 to the list of applicable products.	Y	Content update.

6 Index