

[MS-RSWSSFA]: Report Server Web Service Specification for SharePoint Forms Authentication: ReportServiceAuthentication

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard

specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

| Date | Revision History | Revision Class | Comments |
|------------|------------------|----------------|---|
| 08/07/2009 | 0.1 | Major | First release. |
| 11/06/2009 | 0.1.1 | Editorial | Revised and edited the technical content. |
| 03/05/2010 | 0.2 | Minor | Updated the technical content. |
| 04/21/2010 | 0.2.1 | Editorial | Revised and edited the technical content. |
| 06/04/2010 | 0.3 | Minor | Updated the technical content. |

Contents

| | |
|--|-----------|
| 1 Introduction | 4 |
| 1.1 Glossary | 4 |
| 1.2 References..... | 4 |
| 1.2.1 Normative References..... | 4 |
| 1.2.1.1 Prescriptive API References..... | 5 |
| 1.2.2 Informative References | 5 |
| 1.3 Protocol Overview (Synopsis)..... | 5 |
| 1.4 Relationship to Other Protocols..... | 7 |
| 1.5 Prerequisites/Preconditions | 7 |
| 1.6 Applicability Statement..... | 8 |
| 1.7 Versioning and Capability Negotiation..... | 8 |
| 1.8 Vendor-Extensible Fields..... | 8 |
| 1.9 Standards Assignments | 8 |
| 2 Messages | 9 |
| 2.1 Transport..... | 9 |
| 2.2 Messages..... | 9 |
| 2.2.1 Namespaces | 10 |
| 2.2.2 Methods..... | 11 |
| 2.2.3 Types | 11 |
| 2.2.4 SOAP Headers..... | 11 |
| 3 Appendix A: Full WSDL | 12 |
| 4 Appendix B: Product Behavior | 16 |
| 5 Change Tracking | 17 |
| 6 Index | 19 |

1 Introduction

This document specifies an API protocol that is provided by the ReportServiceAuthentication Web service for authenticating users against a report server [\[MSDN-SS2008RS\]](#) when the SharePoint Web application is configured for Forms Authentication.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

SOAP header
SOAP message
WSDL message
WSDL operation
XML namespace

The following terms are specific to this document:

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.

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.

[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>

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

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatte, S., and Winer, D., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[SOAP1.2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.J., Nielsen, H. F., Karmarkar, A., and Lafon, Y., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation 27, April 2007, <http://www.w3.org/TR/soap12-part1/>

[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>

[XMLNS3] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[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/>

1.2.1.1 Prescriptive API References

[MSDN-SS2008RS] Microsoft Corporation, "SQL Server Reporting Services", SQL Server 2008 Books Online, <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

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MSDN-RSAMethods] Microsoft Corporation, "ReportServiceAuthentication Methods", SQL Server 2008 Books Online, <http://msdn.microsoft.com/en-us/library/reportserviceauthentication.methods.aspx>

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

[MSDN-SS2008RS] Microsoft Corporation, "SQL Server Reporting Services", SQL Server 2008 Books Online, <http://msdn.microsoft.com/en-us/library/ms159106.aspx>

[MSDN-SSRSA] Microsoft Corporation, "ReportServiceAuthentication Namespace", SQL Server 2008 Books Online, <http://msdn.microsoft.com/en-us/library/reportserviceauthentication.aspx>

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

1.3 Protocol Overview (Synopsis)

The ReportServiceAuthentication Web service protocol [\[MSDN-SSRSA\]](#) that is described in this document provides methods for retrieving the authentication mode and for authenticating users against a **report server** 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** (see [\[SOAP1.1\]<1>](#)) 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.

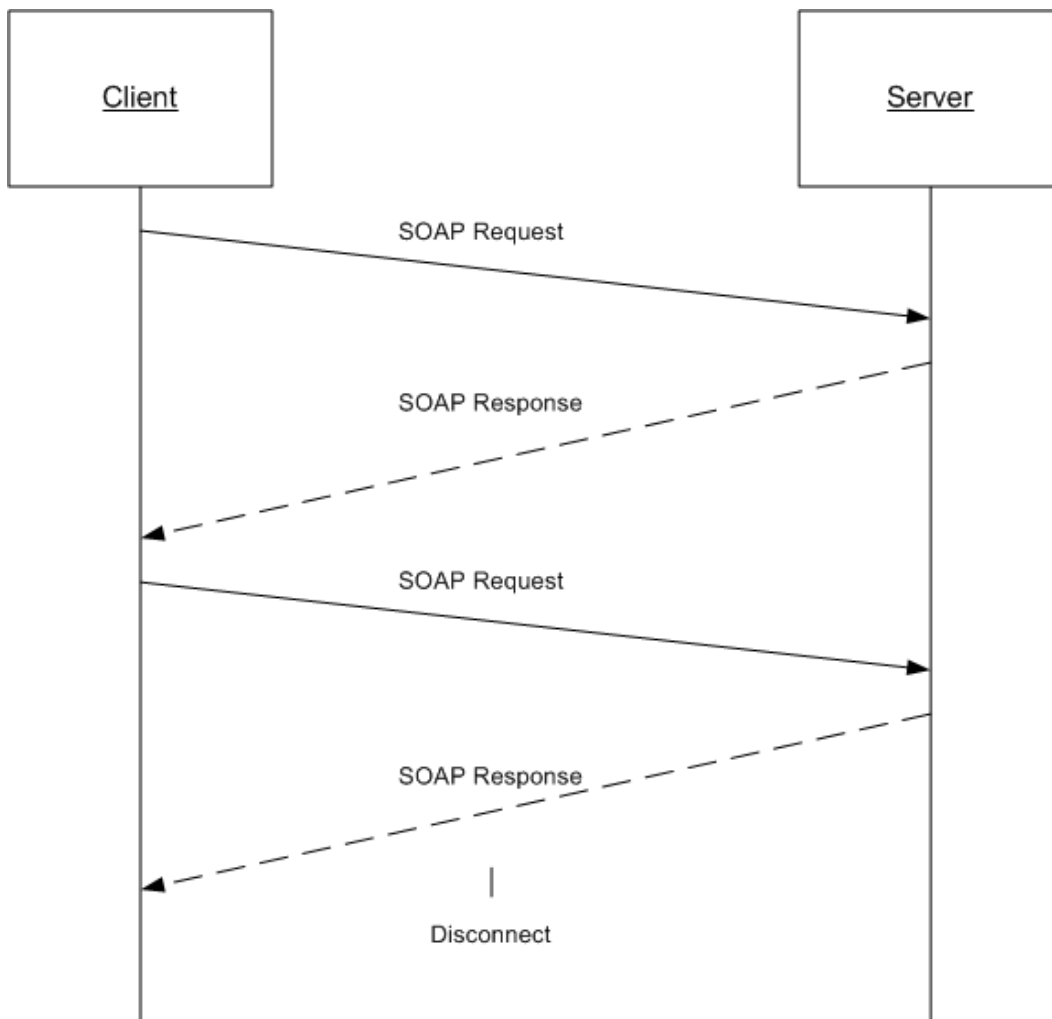


Figure 1: Communication flow for ReportServiceAuthentication

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.

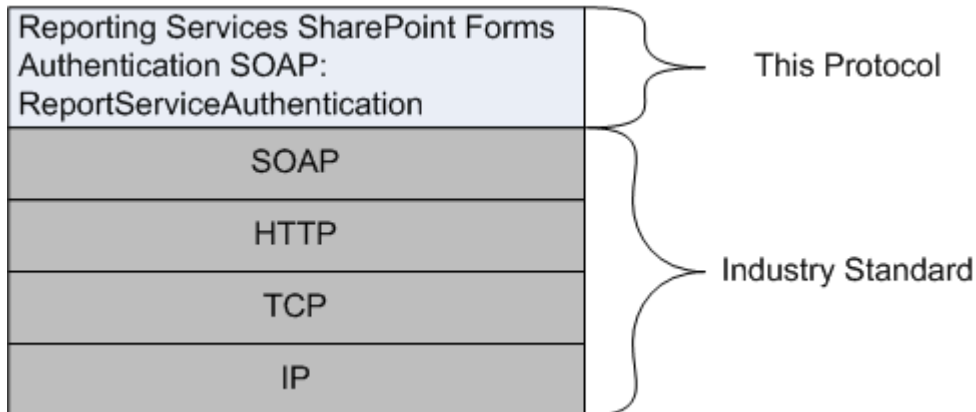


Figure 2: SOAP over HTTP

The ReportServiceAuthentication Web service protocol uses SOAP over HTTPS as shown in the following layering diagram.

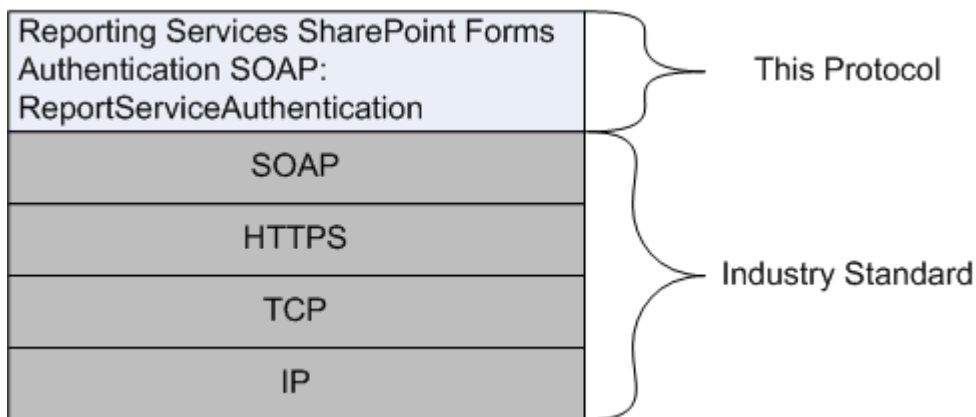


Figure 3: SOAP over HTTPS

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/ReportService2006`

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 [\[SOAP1.1\]<2>](#)

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 [\[MSDN-SSRSA\]](#), 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 [\[XMLNS3\]](#). 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 this document |

| Prefix | Namespace URI | Reference |
|--------|---|------------------------------|
| s | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1] |
| 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 [\[MSDN-RSAMethods\]](#).

| 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 [\[MSDN-SSRSA\]](#).

| 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>

```

```

    </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>
    <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 product versions:

- Microsoft® SQL Server® 2005 Service Pack 2 (SP2)
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

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 1.3](#): SQL Server 2008 Reporting Services and SQL Server 2008 R2 Reporting Services support SOAP 1.2 [\[SOAP1.2\]](#) binding.

[<2> Section 2.1](#): SQL Server 2008 Reporting Services and SQL Server 2008 R2 Reporting Services support SOAP 1.2 [\[SOAP1.2\]](#) binding.

5 Change Tracking

This section identifies changes made to [MS-RSWSSFA] protocol documentation between April 2010 and June 2010 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

Major and minor changes can be described further using the following revision types:

- New content added.
- Content update.
- 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.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision 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.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Revision Type |
|------------------------------|---|------------------------------|----------------------|
| 2.2 Messages | 429961 Removed DateTime mapping to time. DateTime does not map to time; it maps to dateTime. | N | Content removed. |

6 Index

A

authentication ([section 1.3](#) 5, [section 1.5](#) 7)
[AuthenticationMode type](#) 11

C

[Change tracking](#) 17
[classes](#) 9

D

[data types](#) 9

E

[enumerations](#) 9

F

[Forms Authentication](#) 5

G

[GetAuthenticationMode method](#) 11
[Glossary](#) 4

I

[informative references](#) 5
[inherited classes](#) 9

L

[Logoff method](#) 11
[LogonUser method](#) 11

M

[mapping data types](#) 9
[message format](#) 9
[messages](#) 9
methods ([section 2.2](#) 9, [section 2.2.2](#) 11)

N

[Namespaces](#) 10
[normative references](#) 4

O

[out parameters](#) 9

P

[prescriptive API references](#) 5
[protocol message format](#) 9
[protocol server endpoint](#) 7

[protocol servers](#) 9
[protocols](#) 7
[public properties](#) 9

R

references
[informative](#) 5
[normative](#) 4
[prescriptive](#) 5
[report server](#) 5
ReportServiceAuthentication
[about](#) 5
authentication ([section 1.3](#) 5, [section 1.5](#) 7)
[classes](#) 9
[data type mapping](#) 9
[messages](#) 9
methods ([section 2.2](#) 9, [section 2.2.2](#) 11)
[protocol message format](#) 9
[protocol server endpoint](#) 7
[protocols](#) 7
[security](#) 5
supported transports ([section 1.7](#) 8, [section 2.1](#) 9)
[types](#) 11

S

[SOAP headers](#) 9
[SOAP over HTTP](#) 7
[SOAP over HTTPS](#) 7
[SOAP 1.2 binding](#) 16

T

[Tracking changes](#) 17
[type definitions](#) 11

W

[WSDL](#) 12
[WSDL XML Schema types](#) 9

X

[XML namespaces](#) 10