

[MS-SSNWS-Diff]:

Native Web Services Protocol

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~~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 [Open Specification Promise](#) or the [Community Promise](#). 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 jplq@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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, ~~email~~ email 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
08/078/7 /2009	0.1	Major	First release.
11/ 066 /2009	0.1.1	Editorial	Revised Changed language and edited formatting in the technical content.
03/053/5 /2010	0.2	Minor	Updated Clarified the meaning of the technical content.
044/21 /2010	0.2.1	Editorial	Revised Changed language and edited formatting in the technical content.
06/046/4 /2010	0.3	Minor	Updated Clarified the meaning of the technical content.
09/039/3 /2010	0.3.1	Editorial	Changed language and formatting in the technical content.
02/092/9 /2011	0.4	Minor	Clarified the meaning of the technical content.
07/077/7 /2011	0.4	No change None	No changes to the meaning, language, or formatting of the technical content.
11/ 033 /2011	0.4	No change None	No changes to the meaning, language, or formatting of the technical content.
011/19 /2012	0.4	No change None	No changes to the meaning, language, or formatting of the technical content.
022/23 /2012	0.4	No change None	No changes to the meaning, language, or formatting of the technical content.
033/27 /2012	0.4	No change None	No changes to the meaning, language, or formatting of the technical content.
055/24 /2012	0.4	No change None	No changes to the meaning, language, or formatting of the technical content.
066/29 /2012	1.0	Major	Significantly changed Updated and revised the technical content.
077/16 /2012	1.0	No change None	No changes to the meaning, language, or formatting of the technical content.
10/ 088 /2012	1.0	No change None	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	1.0	No change None	No changes to the meaning, language, or formatting of the technical content.
033/26 /2013	1.0	No change None	No changes to the meaning, language, or formatting of the technical content.
066/11 /2013	1.0	No change None	No changes to the meaning, language, or formatting of the technical content.
08/088/8 /2013	1.0	No	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
		change None	the technical content.
12/ 055 /2013	2.0	Major	Significantly changed Updated and revised the technical content.
022 /11/2014	2.0	No change None	No changes to the meaning, language, or formatting of the technical content.
055 /20/2014	2.0	No change None	No changes to the meaning, language, or formatting of the technical content.
<u>5/10/2016</u>	<u>2.0</u>	<u>None</u>	<u>No changes to the meaning, language, or formatting of the technical content.</u>

Table of Contents

1	Introduction	7
1.1	Glossary.....	7
	References	9
1.2	9	
1.2.1	Normative References	10
1.2.2	Informative References	11
1.3	Overview.....	11
1.4	Relationship to Other Protocols	12
1.5	Prerequisites/Preconditions.....	13
1.6	Applicability Statement	14
1.7	Versioning and Capability Negotiation	14
1.8	Vendor-Extensible Fields	14
1.9	Standards Assignments.....	14
2	Messages.....	15
2.1	Transport.....	15
2.2	Common Message Syntax	15
2.2.1	Namespaces.....	15
2.2.2	Messages.....	16
2.2.2.1	sqlbatchSoapIn.....	16
2.2.2.1.1	sqlbatchSoapIn SOAP Body.....	16
2.2.2.1.2	sqlbatchSoapIn SOAP Headers	17
2.2.2.1.2.1	applicationName SOAP Header.....	20
2.2.2.1.2.2	clientInterface SOAP Header.....	20
2.2.2.1.2.3	clientNetworkID SOAP Header	20
2.2.2.1.2.4	clientPID SOAP Header	20
2.2.2.1.2.5	environmentChangeNotifications SOAP Header	21
2.2.2.1.2.6	hostName SOAP Header.....	22
2.2.2.1.2.7	initialDatabase SOAP Header	22
2.2.2.1.2.8	initialLanguage SOAP Header.....	22
2.2.2.1.2.9	notificationRequest SOAP Header	23
2.2.2.1.2.10	sqlSession SOAP Header	23
2.2.2.2	sqlbatchSoapOut.....	24
2.2.2.2.1	sqlbatchSoapOut SOAP Body.....	24
2.2.2.2.1.1	sqlbatchResult.....	24
2.2.2.2.1.1.1	sqlbatchResult.SqlRowSet.....	25
2.2.2.2.1.1.2	sqlbatchResult.SqlXml.....	26
2.2.2.2.1.1.3	sqlbatchResult.SqlMessage	27
2.2.2.2.1.1.4	sqlbatchResult.SqlRowCount	28
2.2.2.2.1.1.5	sqlbatchResult.SqlResultCode	28
2.2.2.2.1.1.6	sqlbatchResult.SqlTransaction	28
2.2.2.2.2	sqlbatchSoapOut SOAP Header.....	29
2.2.2.2.2.1	sqlSession SOAP Header	30
2.2.3	Elements	30
2.2.4	Complex Types.....	31
2.2.4.1	ArrayOfSqlParameter.....	31
2.2.4.1.1	SqlParameter	32
2.2.4.1.2	SqlParameter.Value	34
2.2.5	Simple Types	34
2.2.5.1	sqlCompareOptionsList	35

2.2.5.2	sqlTypes	36
2.2.5.3	sqlDbTypeEnum	39
2.2.6	Attributes	41
2.2.7	Groups	41
2.2.8	Attribute Groups	41
2.3	Directory Service Schema Elements	41
3	Protocol Details	42
3.1	Batch_EPSoap Server Details	42
3.1.1	Abstract Data Model	42
3.1.1.1	Session-specific Structures	42
3.1.2	Timers	42
3.1.3	Initialization	42
3.1.4	Message Processing Events and Sequencing Rules	43
3.1.4.1	Single sqlbatch	43
3.1.4.1.1	Messages	43
3.1.4.1.1.1	sqlbatchSoapIn	44
3.1.4.1.1.2	sqlbatchSoapOut	44
3.1.4.1.2	Elements	44
3.1.4.1.2.1	sqlbatch	44
3.1.4.1.2.2	sqlbatchResponse	44
3.1.4.1.3	Complex Types	44
3.1.4.1.3.1	SqlResultStream	44
3.1.4.1.3.2	ArrayOfSqlParameter	44
3.1.4.1.4	Simple Types	44
3.1.4.1.4.1	ParameterDirection	45
3.1.4.1.5	Attributes	45
3.1.4.1.6	Groups	45
3.1.4.1.7	Attribute Groups	45
3.1.4.2	Session-based sqlbatch	46
3.1.4.2.1	Messages	47
3.1.4.2.1.1	sqlbatchSoapIn	47
3.1.4.2.1.2	sqlbatchSoapOut	47
3.1.4.2.2	Elements	47
3.1.4.2.2.1	sqlSession	47
3.1.4.2.2.2	sqlbatch	48
3.1.4.2.2.3	sqlbatchResponse	48
3.1.4.2.3	Complex Types	48
3.1.4.2.3.1	SqlResultStream	48
3.1.4.2.3.2	ArrayOfSqlParameter	48
3.1.4.2.4	Simple Types	48
3.1.4.2.4.1	ParameterDirection	48
3.1.4.2.5	Attributes	48
3.1.4.2.6	Groups	48
3.1.4.2.7	Attribute Groups	49
3.1.4.3	Authentication	49
3.1.5	Timer Events	49
3.1.6	Other Local Events	49
4	Protocol Examples	50
4.1	SOAP Requests	50
4.1.1	SOAP Request with No Parameters	50
4.1.2	SOAP Request with SOAPAction Header	50
4.1.3	SOAP Request with Parameters	50

4.1.4	SOAP Request with Additional Parameter Attributes	51
4.1.5	SOAP Request with sqlSession.initiate	51
4.1.6	SOAP Request with sqlSession.sessionId	52
4.1.7	SOAP Request with sqlSession.terminate	52
4.2	SOAP Responses	53
4.2.1	SOAP Response with No Output Parameters	53
4.2.2	SOAP Response with Output Parameters.....	54
4.2.3	SOAP Response with Additional Output Parameter Attributes.....	56
4.2.4	SOAP Response to a Request with sqlSession.initiate	58
4.2.5	SOAP Response to a Request with sqlSession.sessionId	58
4.2.6	SOAP Response to a Request with sqlSession.terminate	60
4.2.7	SOAP Fault Response	61
5	Security	63
5.1	Security Considerations for Implementers	63
5.2	Index of Security Parameters.....	63
6	Appendix A: Full WSDL	64
7	Appendix B: Product Behavior	75
8	Change Tracking	76
9	Index.....	77

1 ~~1~~ Introduction

The Native Web Services (NWS) protocol is an application layer request/response protocol that facilitates interaction with a database server and provides for specification of requests to SQL Server and the return of data. This protocol is a specific **web services** implementation that uses the standard **Simple Object Access Protocol (SOAP)**.

Sections ~~1.5, 1.8, 1.9, 2,~~ and 3 of this specification are normative ~~and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms.~~ All other sections and examples in this specification are informative.

1.1 ~~1.1~~ Glossary

~~The~~ This document uses the following terms ~~are defined in [MS-GLOS]:~~

~~certificate
globally unique identifier (GUID)
SOAP action
SOAP body
SOAP fault
SOAP header
SOAP message
SOAP mustUnderstand attribute
UTF-16
UTF-8
Web Services Description Language (WSDL)
web services
WSDL message
WSDL operation
WSDL port type
XML
XML namespace
XML schema (XSD)~~

~~The following terms are specific to this document:~~

~~**attribute:** A characteristic of some object or entity, typically encoded as a name/value pair.~~

~~**authentication:** The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.~~

~~**Basic:** An authentication access type supported by HTTP as defined by [RFC2617].~~

~~**Digest:** An authentication access type supported by HTTP as defined by [RFC2617].~~

~~**certificate:** A certificate is a collection of **attributes** and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for **authentication** and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.~~

Digest Access Authentication: A mechanism built on top of HTTP that allows a client program to provide credentials without having to send a user name and password in plaintext when making a request. For more information, see [RFC2617].

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Kerberos: An authentication access type as defined by [RFC1964].

Negotiate: An authentication access type supported by HTTP as defined by [RFC4559].

~~**NTLM:** An authentication access type as defined by [MSDN-NTLM].~~

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

NWS object: An instance of the Native Web Services (NWS) protocol that is created by receiving a sqlbatch request.

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

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a **SOAP message** to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP fault: A container for error and status information within a **SOAP message**. See [SOAP1.2-1/2007] section 5.4 for more information.

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.

SOAP mustUnderstand attribute: A global, Boolean **attribute** that is used to indicate whether a header entry is mandatory or optional for the recipient to process. See [SOAP1.2-1/2007] section 5.2.3 for more information.

UTF-16: A standard for encoding Unicode characters, defined in the Unicode standard, in which the most commonly used characters are defined as double-byte characters. Unless specified otherwise, this term refers to the UTF-16 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

UTF-8: A byte-oriented standard for encoding Unicode characters, defined in the Unicode standard. Unless specified otherwise, this term refers to the UTF-8 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

web service: A software system designed to support interoperable machine-to-machine interaction over a network, using XML-based standards and open transport protocols.

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

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.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [XML1.0].

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

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2—References

1.2 ~~References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.~~

1.2.1—~~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>

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., et al., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, <http://www.rfc-editor.org/rfc/rfc2617.txt>

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

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and Ingersoll, W., "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, October 2005, <http://www.rfc-editor.org/rfc/rfc4178.txt>

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

[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/>

[SOAP1.2-1/2003] Gudgin, M., Hadley, M., Mendelsohn, N., et al., ~~Eds.,~~ "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[SOAP1.2-2/2003] Gudgin, M., Hadley, M., Mendelsohn, N., et al., ~~Eds.,~~ "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>

~~[WSSE-UsernameToken] See [WSSUTP].~~

[WSSUTP] OASIS ~~Standard,~~ "Web Services Security UsernameToken Profile 1.0", ~~OASIS Standard,~~ March 2004, <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>

~~[XML10] See [XML10/5].~~

[XML10/5] Bray, T., Paoli, J., Sperberg-McQueen, C.M., et al., Eds., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>

[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/>

~~[XMLNS3] See [XMLNS].~~

[XMLSCHEMA1] Thompson, H.S., 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., [EdsEd.](#), "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 ~~1.2.2~~ Informative References

~~[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary":~~

[MSDN-DEES] Microsoft Corporation, "Database Engine Error Severities", <http://msdn.microsoft.com/en-us/library/ms164086.aspx>

~~[MSDN-NTLM] Microsoft Corporation, "Microsoft NTLM", <http://msdn.microsoft.com/en-us/library/aa378749.aspx>~~

[MSDN-SQLCollation] Microsoft Corporation, "Selecting a SQL Server Collation", <http://msdn.microsoft.com/en-us/library/ms144250.aspx>

[MSDN-SSLNXWS] Microsoft Corporation, "Setting the Server to Listen for Native XML Web Services Requests", <http://msdn.microsoft.com/en-us/library/ms191310.aspx>

[MSDN-TSQL] Microsoft Corporation, "Transact-SQL Overview", [http://msdn.microsoft.com/en-us/library/aa260642\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa260642(SQL.80).aspx)

[MSDN-XMLSNET] Microsoft Corporation, "XML Serialization in the .NET Framework", <http://msdn.microsoft.com/en-us/library/ms950721.aspx>

~~[NTLM] See [MSDN-NTLM].~~

~~[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996, <http://www.rfc-editor.org/rfc/rfc1964.txt>~~

1.3 ~~1.3~~ Overview

The Native Web Services Protocol is an application-level protocol that is used to transfer requests and responses between clients and database server systems. In such systems, the client will typically establish a connection with the server. Once the connection is established using the HTTP ([RFC2616]) or HTTPS ([RFC2818]) protocol, **SOAP messages**, SOAP1.1 ([SOAP1.1]) or SOAP1.2 ([SOAP1.2-1/2003], [SOAP1.2-2/2003]), are used to communicate between the client and the server.

The NWS protocol uses the security facilities built into HTTP or HTTPS for **authentication** and identification and also for channel encryption negotiation. The protocol uses the facilities built into SOAP for specification of requests from client to server (including Transact SQL queries; for more information, see [MSDN-TSQL]) and for returning data from server to client. The following diagram depicts a (simplified) typical flow of communication in the protocol.

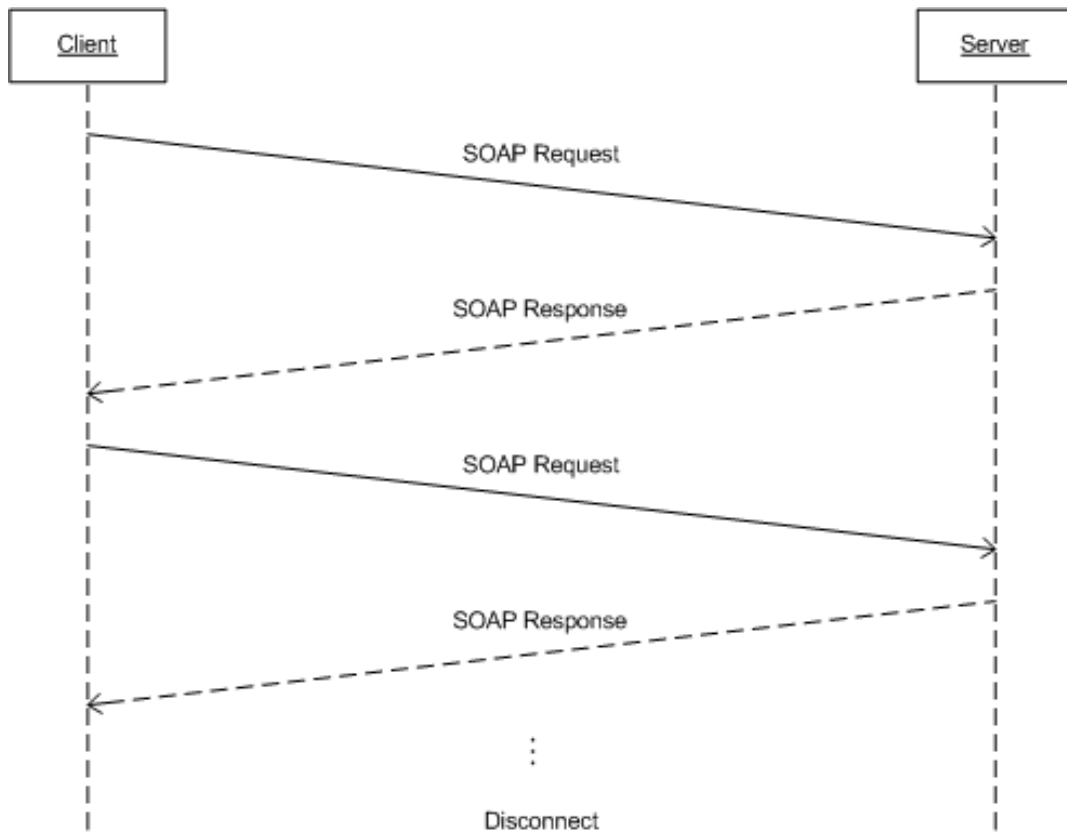


Figure 1.4: Communication flow in the Native Web Services protocol

The following example is a high-level description of the messages exchanged between the client and the server to execute a simple client request such as the execution of an [MSDN-TSQL] statement. It is assumed that the client and the server have already established a connection and authentication has succeeded.

```
Client:SOAP sqlbatch
```

The server executes the statement and then sends back the results to the client.

```
Server:SOAP sqlbatchResponse
```

1.4 ~~1.4~~ Relationship to Other Protocols

The NWS protocol uses SOAP over HTTP as shown in the following layering diagram.

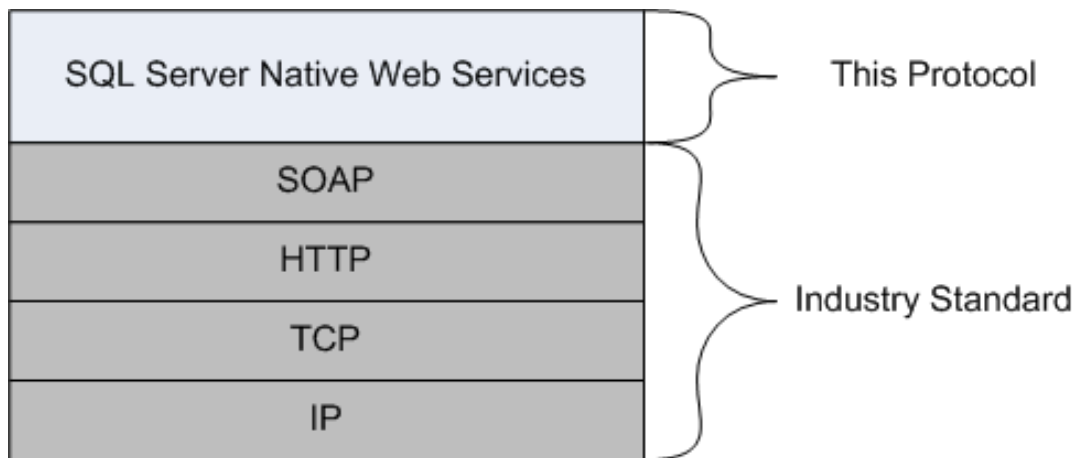


Figure 22: SOAP over HTTP

The protocol depends on the underlying network stacks being established prior to communications with NWS.

The NWS protocol uses SOAP over HTTPS for network encryption as shown in the following layering diagram.

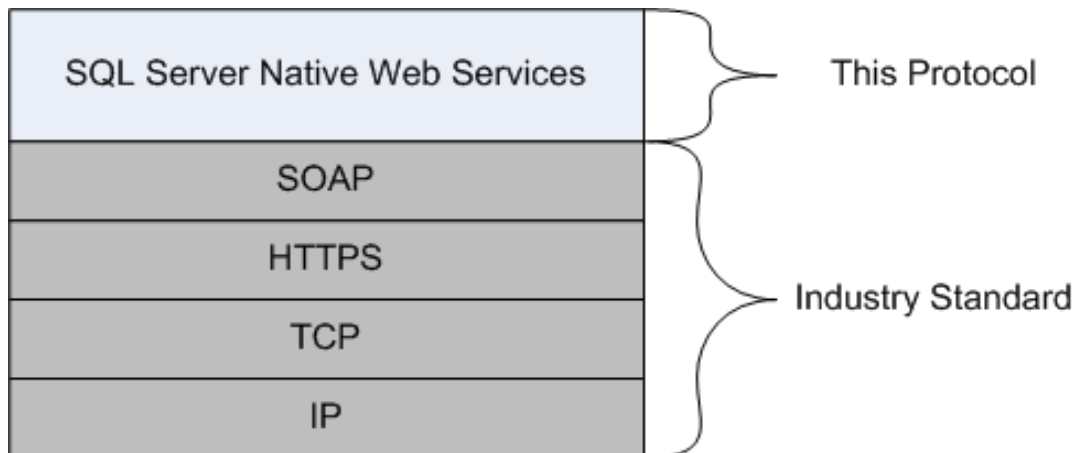


Figure 33: SOAP over HTTPS

1.5 ~~1.5~~ Prerequisites/Preconditions

It is assumed that the client has already discovered the server and established a network transport connection for use with NWS.

No security association is assumed to have been established at the lower layer before NWS begins functioning. For [RFC4178] or [RFC2617] authentication to be used, [RFC4178] or [RFC2617] support must be available on both the client and server machines. If channel encryption is to be used, [RFC2818] support must be present on both the client and server machines, and a **certificate** that is suitable for encryption must be deployed on the server machine.

1.6 ~~1.6~~—Applicability Statement

The NWS protocol is appropriate for use to facilitate request/response communications between an application and a database server in web services application scenarios where network or local connectivity is available.

1.7 ~~1.7~~—Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- ~~1.7.1~~ **Supported Transports:** This protocol uses multiple transports with SOAP as described in section 2.1.
- ~~1.7.2~~ **Protocol Versions:** This protocol has only one version and has only one **WSDL port type** version with a single operation. The use of the operation is described in section 3.1.
- ~~1.7.3~~ **Security and Authentication Methods:** This protocol supports the following authentication methods: **Negotiate**, **NTLM**, **Kerberos**, **Digest Access**, and **Basic**.
- ~~1.7.4~~ **Localization:** This protocol includes text strings in various messages. This protocol uses **UTF-8** and **UTF-16** encoded strings.
- ~~1.7.5~~ **Capability Negotiation:** This protocol does not support negotiation of the interface version to use. Instead, an implementation must be configured with the interface version to use, as described in the following paragraph.

The NWS protocol does not provide facilities for capability negotiation; it is fixed. Depending on the configuration of the server, the client can request which authentication type to use; whether to use SOAP1.1 or SOAP1.2; and whether to use HTTP or HTTPS. Ultimately, the server decides whether the SOAP message sent by the client meets the server requirements.

1.8 ~~1.8~~—Vendor-Extensible Fields

None.

1.9 ~~1.9~~—Standards Assignments

Parameter	Value	Reference
None		

2 ~~2~~—Messages

~~2.1~~—Transport

2.1 Transport

The NWS protocol supports both SOAP v1.1 and SOAP v1.2 requests. The corresponding response MUST be sent in the same SOAP version as the request. As mentioned in section 1.4, the communication is only supported over the HTTP/HTTPS protocol, specifically HTTP v1.1. If a request is sent using HTTP v1.0, the server MUST reject the request and return an *HTTP 505 Version not supported* error.

~~2.1.2.2.2~~—Common Message Syntax

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

This protocol follows the standard **XML** data usage as specified by [SOAP1.1], [SOAP1.2-1/2003], [SOAP1.2-2/2003], and [XML10/5]. Following the rules specified by these standards, certain elements allow for the existence of unknown **attributes**. These unknown attributes are ignored unless otherwise explicitly specified within this document.

~~2.2.1~~—Namespaces

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix to 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
sql	"http://schemas.microsoft.com/sqlserver/2004/SOAP"	None
sqlsoaptypes	"http://schemas.microsoft.com/sqlserver/2004/SOAP/types"	None
sqlrowcount	"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlRowCount"	None
sqlmessage	"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlMessage"	None
sqloptions	"http://schemas.microsoft.com/sqlserver/2004/SOAP/Options"	None
sqlparameter	"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"	None
sqlresultstream	"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlResultStream"	None
sqltransaction	"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlTransaction"	None
sqltypes	"http://schemas.microsoft.com/sqlserver/2004/sqltypes"	None

2.2.2 Messages

2.2.2 Messages

Message	Description
sqlbatchSoapIn	This WSDL message defines the request portion of the SQL Server built-in optional WSDL operation .
sqlbatchSoapOut	This WSDL message defines the response portion of the SQL Server built-in optional WSDL operation.

2.2.2.1 sqlbatchSoapIn

2.2.2.1 sqlbatchSoapIn

The sqlbatchSoapIn WSDL message is a SOAP request from the client to the server. It has an optional **SOAP action** value of "http://schemas.microsoft.com/sqlserver/2004/SOAPsqlbatch". The following set of XML namespaces is used throughout the subsections under this section:

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:sqlparameter=
```

```
"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"
```

In addition to the required **SOAP body**, the sqlbatchSoapIn message also supports optional **SOAP headers**. The set of optional SOAP headers allowed includes the [WSSUTP] and the set of SOAP headers defined in the "http://schemas.microsoft.com/sqlserver/2004/SOAP/Options" namespace. The SOAP headers are specified in section 2.2.2.1.2.

~~2.1.1.1~~ ~~2.2.2.1.1~~ ~~2.2.2.1.1~~ sqlbatchSoapIn SOAP Body

The following describes the element within the SOAP request body under the "http://schemas.microsoft.com/sqlserver/2004/SOAP" namespace.

```
<xsd:element name="sqlbatch">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element minOccurs="1" maxOccurs="1"  
name="BatchCommands" type="xsd:string" />  
      <xsd:element minOccurs="0" maxOccurs="1"  
name="Parameters"  
type="sqlparameter:ArrayOfSqlParameter"  
nillable="true" />  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

sqlbatch.BatchCommands: This required element holds the string that makes up the Transact-SQL query to be executed. This element **MUST** exist in the request. For more information about Transact-SQL syntax, see [MSDN-TSQL].

sqlbatch.Parameters: This optional element is a complex type that defines the list of parameters associated with the query syntax specified by the **BatchCommands** element. The client application may send this element as part of the request. The details of this element are defined by the **ArrayOfSqlParameter** complex type, which is described in section 2.2.4.1.

~~2.1.1.1.2.2.1.2~~ ~~2.2.2.1.2~~ sqlbatchSoapIn SOAP Headers

The **sqlbatchSoapIn** SOAP headers are defined by the "http://schemas.microsoft.com/sqlserver/2004/SOAP/Options" namespace.

```
<xsd:element name="applicationName">
  <xsd:annotation>
    <xsd:documentation>Set the application name for the
login.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:string"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The application name
to set for the login.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="clientInterface">
  <xsd:annotation>
    <xsd:documentation>Set the client interface
for the login.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:string"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The client interface
to set for the login.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="clientNetworkID">
  <xsd:annotation>
    <xsd:documentation>Set the client network ID
for the login.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:base64Binary"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The client network ID to
set for the login.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="clientPID">
  <xsd:annotation>
    <xsd:documentation>Set the client process ID
for the login.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:long"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The client process ID to
set for the login.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="environmentChangeNotifications">
  <xsd:annotation>
    <xsd:documentation>Receive environment change notifications.</xsd:documentation>
```

```

</xsd:annotation>
<xsd:complexType>
  <xsd:attribute name="databaseChange"
default="false" type="xsd:boolean" form="unqualified">
  <xsd:annotation>
    <xsd:documentation>Receive notifications of database
changes.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
  <xsd:attribute name="languageChange" default="false"
type="xsd:boolean" form="unqualified">
  <xsd:annotation>
    <xsd:documentation>Receive notifications of
language changes.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
  <xsd:attribute name="transactionBoundary" default="false" type="xsd:boolean"
form="unqualified">
  <xsd:annotation>
    <xsd:documentation>Receive notifications of transaction
boundaries.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="hostName">
  <xsd:annotation>
    <xsd:documentation>Set the host name for the
login.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:string"
form="unqualified" use="required">
    <xsd:annotation>
      <xsd:documentation>The host name to set for the
login.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="initialDatabase">
  <xsd:annotation>
    <xsd:documentation>Set initial database on
login.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:string"
form="unqualified" use="required">
    <xsd:annotation>
      <xsd:documentation>The name of the initial
database to attach to.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="optional" default="false"
type="xsd:boolean" form="unqualified">
  <xsd:annotation>
    <xsd:documentation>Whether the initial database
is optional or not.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
  <xsd:attribute name="filename" type="xsd:string"
form="unqualified">
  <xsd:annotation>
    <xsd:documentation>The filename of the database
to attach to.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>

```

```

<xsd:element name="initialLanguage">
  <xsd:annotation>
    <xsd:documentation>Set initial language
to set.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="value" type="xsd:string"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The name of the initial
language to set.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="optional" default="false"
type="xsd:boolean" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>Whether the initial language
is optional or not.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="notificationRequest">
  <xsd:annotation>
    <xsd:documentation>Requests query notifications
for the request.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="notificationId" type="xsd:string"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The notification
identifier.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="deliveryService" type="xsd:string"
form="unqualified" use="required">
      <xsd:annotation>
        <xsd:documentation>The delivery
service.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="timeout" type="xsd:integer"
form="unqualified">
      <xsd:annotation>
        <xsd:documentation>The timeout
value.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="sqlSession">
  <xsd:annotation>
    <xsd:documentation>SQL Server SOAP
Session</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="initiate" default="false"
type="xsd:boolean" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>Set to 'true' to request
to start a new session.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="terminate" default="false"
type="xsd:boolean" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>Set to 'true' to request to
terminate an existing session.</xsd:documentation>

```

```

        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="sessionId" type="xsd:base64Binary"
form="unqualified">
        <xsd:annotation>
          <xsd:documentation>The ID of a
session.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="timeout" type="xsd:int"
form="unqualified">
        <xsd:annotation>
          <xsd:documentation>The timeout in seconds before
the session expires.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="transactionDescriptor"
type="xsd:base64Binary" form="unqualified">
        <xsd:annotation>
          <xsd:documentation>The descriptor of a transaction
to enlist to.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>

```

Each of the SOAP header elements supports the standard **SOAP mustUnderstand attribute** and the **actor** attribute. If the request is created by using the SOAP 1.2 format, each of the SOAP header elements also supports the standard SOAP **relay** attribute. The details of each SOAP header are described in their corresponding subsection, 2.2.2.1.2.1 through 2.2.2.1.2.10.

~~2.1.1.1.2.1~~ ~~2.2.1.2.1~~ **applicationName SOAP Header**

applicationName: This optional element permits the client application to specify the name of the client connecting to the server. Specifying a value helps to identify which connection is established by which client application when troubleshooting the server. The value is specified by using the required value attribute.

applicationName.value: This required attribute of string type specifies the name of the client application that is connecting to the server.

~~2.1.1.1.2.2~~ ~~2.2.1.2.2~~ **clientInterface SOAP Header**

clientInterface: This optional element permits the client application to specify the name of the API used by the client application. Specifying a value helps to identify the scenario when troubleshooting the server. The value is specified by using the required value attribute.

clientInterface.value: This required attribute of string type specifies the name of the API the client application is using.

~~2.1.1.1.2.3~~ ~~2.2.1.2.3~~ **clientNetworkID SOAP Header**

clientNetworkID: This optional element permits the client application to specify the identification number of the network card the client application used to connect to the server. Specifying a value helps to identify the scenario when troubleshooting the server. The value is specified by using the required value attribute.

clientNetworkID.value: This required attribute of XML base64 binary type specifies the identification number.

~~2.1.1.1.2.4~~ ~~2.2.1.2.4~~ **clientPID SOAP Header**

clientPID: This optional element permits the client application to specify the process identification number of the client application. Specifying a value helps to troubleshoot the exact client application that is having problems. The value is specified by using the required value attribute.

clientPID.value: This required attribute of XML long type specifies the identification number of the client application process.

~~2.1.1.1.2.5~~ ~~2.2.1.2.5~~ ~~2.2.2.1.2.5~~ **environmentChangeNotifications SOAP Header**

environmentChangeNotifications: This optional element permits the client application, by using the optional attributes associated with this element, to specify the set of database session environment change notifications the client application requests to receive.

environmentChangeNotifications.databaseChange: This optional attribute of XML Boolean type specifies whether the client application requests to receive notification for database usage changes within the current database session. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The server MUST send a notification to the client if, in the current database session, the database context has changed. The notification is delivered to the client as an element of type sqlbatchResponse.sqlbatchResult.SqlMessage as part of the server response. For details, see section 2.2.2.2.1.1.3.
false	The server MUST NOT send a notification to the client if, in the current database session, the database context has changed.

environmentChangeNotifications.languageChange: This optional attribute of XML Boolean type specifies whether the client application requests to receive notification for language setting changes within the current database session. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The server MUST send a notification to the client if, in the current database session, the language environment has changed. The notification is delivered to the client as an element of type sqlbatchResponse.sqlbatchResult.SqlMessage as part of the server response. For details, see section 2.2.2.2.1.1.3.
false	The server MUST NOT send a notification to the client if, in the current database session, the language environment has changed.

environmentChangeNotifications.transactionBoundary: This optional attribute of XML Boolean type specifies whether the client applications want to receive notification for transaction changes within the current database session. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The server MUST send a notification to the client if, in the current database session, the transaction context has changed. The notification is delivered to the client as an element of type sqlbatchResponse.sqlbatchResult.SqlTransaction as part of the server response. For details, see section 2.2.2.2.1.1.6.
false	The server MUST NOT send a notification to the client if, in the current database session, the transaction context has changed.

~~2.1.1.1.2.6~~~~2.2.1.2.6~~ ~~2.2.2.1.2.6~~ **hostName SOAP Header**

hostName: This optional element permits the client application to specify the host name of the client machine. Specifying a value helps to troubleshoot the exact client application that is having problems. The value is specified by using the required value attribute.

hostName.value: This required attribute of string type specifies the value of the client host name.

~~2.1.1.1.2.7~~~~2.2.1.2.7~~ ~~2.2.2.1.2.7~~ **initialDatabase SOAP Header**

initialDatabase: This optional element provides the ability for the client application to specify the name of the database to initially log in to. The value is specified by using the required value attribute.

initialDatabase.value: This required attribute of string type specifies the name of the database to initially log in to.

initialDatabase.optional: This optional attribute of Boolean type describes whether the specified initial database exists and whether the client application user account can successfully log in to that database. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The database specified by the value attribute MUST exist or the database file specified by the filename attribute MUST successfully attach and the user MUST successfully log in to the specified database. If any of the above fails, then the server MUST return an error and terminate the connection.
false	The database specified by the value attribute SHOULD be used to attempt the login. If the database specified does not exist or the user does not have login permissions for the specified database, then the <code>user's user's</code> default login database can be used to attempt the login.

initialDatabase.filename: This optional attribute of string type describes the file path and file name of the database to attach as part of the login.

~~2.1.1.1.2.8~~~~2.2.1.2.8~~ ~~2.2.2.1.2.8~~ **initialLanguage SOAP Header**

initialLanguage: This optional element describes the name of the language to set the login to. The value is specified by using the required value attribute.

initialLanguage.value: This required attribute of string type describes the name of the language to set as part of login.

initialLanguage.optional: This optional attribute of Boolean type describes whether the specified initial language will succeed. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The language specified by the value attribute SHOULD be used to set the language setting of the connection. If the server cannot be set to the language specified by the initialLanguage.value attribute, the server will use the default login language of the user.
false	The language specified by the value attribute MUST exist. If the specified language cannot be set, the server MUST return a SOAP fault message error and terminate the connection.

~~2.1.1.1.2.9~~2.2.1.2.9 ~~2.2.2.1.2.9~~ notificationRequest SOAP Header

notificationRequest: This optional element provides the ability for the client application to specify the notification service that the server MUST use to send query notifications to the client. The ID and service name are specified by using the required **notificationId** and **deliveryService** attributes.

notificationRequest.notificationId: This required attribute of string type specifies the ID that is associated with the notification request.

notificationRequest.deliveryService: This required attribute of string type specifies the name of the Service Broker service that is listening for query notifications.

notificationRequest.timeout: This optional attribute of XML integer type describes the length of time of the notification request in seconds.

~~2.1.1.1.2.10~~2.2.1.2.10 ~~2.2.2.1.2.10~~ sqlSession SOAP Header

sqlSession: This optional element provides the ability for the client to control a named query session. The mechanism to control a named query session is through the use of the following associated optional attributes.

sqlSession.initiate: This optional attribute of Boolean type describes whether the server should create a new session. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The server SHOULD create a new session before executing the query statement in the request.
false	The server MUST NOT create a new named session.

sqlSession.terminate: This optional attribute of Boolean type describes whether the server should terminate the session specified by the **sessionId** attribute. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The server SHOULD terminate the session that is specified by the sessionId attribute at the end of executing the query statement in the request.
false	The server MUST NOT terminate the session that is specified by the sessionId attribute.

sqlSession.sessionId: This optional attribute of XML base64 binary type specifies the session token. This token is first generated by the server when a new session is created. The client may use the same session token that is returned by the server to join a created session to continue executing queries in the same session.

sqlSession.timeout: This optional attribute of XML int type specifies the duration, in seconds, of inactivity in the session before the server terminates the session.

sqlSession.transactionDescriptor: This optional attribute of XML base64 binary type specifies the transaction token. This token is first returned by the server in the **sqlbatchResponse.sqlbatchResult.SqlTransaction** element of the sqlbatchSoapOut message that is described in section 2.2.2.2.1.1.6. The client may use the same transaction token that is returned by the server to join the existing transaction to execute the query statement in the request.

~~2.2.2.2~~ **sqlbatchSoapOut**

2.2.2.2 **sqlbatchSoapOut**

The sqlbatchSoapOut WSDL message is a server response. The following set of XML namespaces is used throughout the subsections under this section:

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sqlresultstream=
"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlResultStream"
xmlns:sqlparameter=
"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/2004/SOAP/types"
xmlns:sqlmessage=
"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlMessage"
xmlns:sqlrowcount=
"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlRowCount"
xmlns:sqltransaction=
"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlTransaction"
```

~~2.1.1.1.3~~2.2.2.2.1 ~~2.2.2.2.1~~ **sqlbatchSoapOut SOAP Body**

The following describes the element within the SOAP response body under the "http://schemas.microsoft.com/sqlserver/2004/SOAP" namespace.

```
<xsd:element name="sqlbatchResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="1"
name="sqlbatchResult" type="sqlresultstream:SqlResultStream"
nillable="false"/>
      <xsd:element minOccurs="0" maxOccurs="1"
name="Parameters" type="sqlparameter:ArrayOfSqlParameter"
nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

sqlbatchResponse.sqlbatchResult: This required element of complex type **SqlResultStream** defines the set of possible XML structures that may be part of the server response to a sqlbatch request. The details of this complex type are defined in section 2.2.2.2.1.1.

sqlbatchResponse.Parameters: This optional element is a complex type that defines the list of output parameters associated with the result of the original sqlbatch request. If there are any corresponding output parameters as a result of the sqlbatch request, the server **MUST** send this element as part of the response. The details of this element are defined in section 2.2.4.1.

~~2.1.1.1.3~~2.2.2.2.1.1 ~~2.2.2.2.1.1~~ **sqlbatchResult**

Referenced by the **sqlbatchResult** element, the **SqlResultStream** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlResultStream" namespace as the following.

```
<xsd:complexType name="SqlResultStream">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SqlRowSet" type="sqlsoaptypes:SqlRowSet" />
    <xsd:element name="SqlXml" type="sqlsoaptypes:SqlXml" />
    <xsd:element name="SqlMessage" type="sqlmessage:SqlMessage" />
    <xsd:element name="SqlRowCount"
type="sqlrowcount:SqlRowCount" />
    <xsd:element name="SqlResultCode"
type="sqlsoaptypes:SqlResultCode" />
    <xsd:element name="SqlTransaction"
type="sqltransaction:SqlTransaction" />
  </xsd:choice>
</xsd:complexType>
```

Each of the subtypes that make up the **SqlResultStream** type is described in a corresponding subsection, 2.2.2.2.1.1.1 through 2.2.2.2.1.1.6.

~~2.1.1.1.3.1.1~~~~2.2.2.2.1.1.1~~ ~~2.2.2.2.1.1.1~~ ~~sqlbatchResult.SqlRowSet~~

sqlbatchResponse.sqlbatchResult.SqlRowSet: This element of complex type **SqlRowSet** describes the portion of the response that represents a resultset. The **SqlRowSet** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types" namespace as the following.

```
<xsd:complexType name="SqlRowSet">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="xsd:schema"/>
    <xsd:any/>
  </xsd:sequence>
</xsd:complexType>
```

This complex type MUST conform to the following rules:

- ~~—~~The complex type MUST have two main components:
 - ~~—~~The first component defined by <xsd:element ref="xsd:schema"/>, hereafter referred to as Schema, can be specified by the server. When specified, it MUST be one or more XML schema elements as defined by [XMLSCHEMA1] and [XMLSCHEMA2], and it MUST contain a valid XML schema.
 - ~~—~~The second component defined by <xsd:any/>, hereafter referred to as DiffGram, MUST be an element named "diffgram" in the following namespace:
"urn:schemas-microsoft-com:xml-diffgram-v1"

The paragraphs that follow define the Schema component and the DiffGram component in more detail. At a basic level, the purpose of these components can be explained as follows:

- ~~—~~The Schema component defines the XML schema for the data representation in the DiffGram component's content. The XML representation of the data in the DiffGram component's content MUST conform to the XML schema defined in the Schema component.
- ~~—~~The DiffGram component encapsulates the values of the data in the resultset.

An example is shown in section 4.2.2.

The server may enable the user to define custom types, simple types or complex types. Based on such definitions, when specifying the XML schema, the server defines custom types, simple types or complex types in the user-defined target namespace; if the user does not specify a namespace, the server defines custom types, simple types or complex types in arbitrary target namespaces, and then references the custom types, simple types or complex types in a subsequent target namespace. The server may also redefine existing simple types or complex types in existing target namespaces, such as "http://schemas.microsoft.com/sqlserver/2004/sqltypes". If the server does redefine the "http://schemas.microsoft.com/sqlserver/2004/sqltypes" namespace, then each type specified MUST have the same XML schema as defined in the original namespace. Whether or not the server specifies custom simple or complex types in one or more namespaces, the server MUST specify a schema in a namespace that defines the element structure of the DiffGram component, hereafter referred to as DataInstance schema.

The DataInstance schema MUST contain exactly one element that will encapsulate the representation of all data in the DiffGram component. This element is referred to as the **RowSet** element. In addition to being a valid XML schema, the DataInstance schema MUST conform to the following rules:

- ~~—~~The **RowSet** element MUST be defined using an anonymous complex type. The complex type MUST be defined to have one child element named **row** with zero minimum occurrence and unbounded maximum occurrence.
- ~~—~~The "http://www.w3.org/2001/XMLSchema" element that defines the **RowSet** element MUST have the **urn:schemas-microsoft-com:xml-msdata:IsDataSet** attribute set to true.
- ~~—~~The "http://www.w3.org/2001/XMLSchema" element that defines the **RowSet** element MUST have the **urn:schemas-microsoft-com:xml-msdata:DataSetName** attribute.
- ~~—~~The "http://www.w3.org/2001/XMLSchema" element that defines the **RowSet** element MUST have the **urn:schemas-microsoft-com:xml-msdata:DataSetNamespace** attribute.
- ~~—~~The **row** element MUST be defined by using an anonymous complex type. The complex type MUST be defined as a sequence of child elements. The sequence MUST match the order of the columns of the resultset. Each child element may be specified with zero minimum occurrence. The names of the child elements are the same as the names of the columns of the resultset. When the resultset columns do not have a name, the names of the elements start at "column1" and the suffix number is incremented for each additional unnamed column.
- ~~—~~Each child element within the **row** element SHOULD be defined in terms of a type defined in the "http://schemas.microsoft.com/sqlserver/2004/sqltypes" namespace. See section 2.2.5.2 for information about the mapping between SQL Server data types and corresponding XML data types. The XML schema may also specify additional properties or restrictions on the simple or complex type definition.

As mentioned in the preceding paragraphs, whether or not the Schema component is specified, the DiffGram component MUST be specified. The DiffGram component MUST have a root element of "<diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">". If the Schema component is specified, the subelements of the root MUST match the sequence of elements as defined by the Schema component.

~~2.1.1.1.3.1.22.2.2.1.1.2 2.2.2.2.1.1.2~~ **sqlbatchResult.SqlXml**

sqlbatchResponse.sqlbatchResult.SqlXml: This element of complex type **SqlXml** describes the portion of the response representing a resultset from a Select for XML query.

```
<xsd:complexType name="SqlXml" mixed="true">
  <xsd:sequence>
    <xsd:any processContents="skip" />
  </xsd:sequence>
</xsd:complexType>
```

```
</xsd:sequence>
</xsd:complexType>
```

If complex type `SqlXml` is present, it describes data that is treated as arbitrary XML data.

~~2.1.1.1.3.1.3~~ ~~2.2.2.1.1.3~~ ~~2.2.2.1.1.3~~ `sqlbatchResult.SqlMessage`

sqlbatchResponse.sqlbatchResult.SqlMessage: This element of complex type **SqlMessage** describes the portion of the response representing a SQL Server message. This includes server-generated error messages, notifications and user-defined messages. The **SqlMessage** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlMessage" namespace as the following.

```
<xsd:complexType name="SqlMessage">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="Class"
      type="sqlmessage:nonNegativeInteger" />
    <xsd:element name="LineNumber"
      type="sqlmessage:nonNegativeInteger" />
    <xsd:element name="Message" type="xsd:string" />
    <xsd:element name="Number"
      type="sqlmessage:nonNegativeInteger" />
    <xsd:element name="Procedure"
      type="xsd:string" minOccurs="0" />
    <xsd:element name="Server"
      type="xsd:string" minOccurs="0" />
    <xsd:element name="Source" type="xsd:string" />
    <xsd:element name="State"
      type="sqlmessage:nonNegativeInteger" />
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="nonNegativeInteger">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="0" />
  </xsd:restriction>
</xsd:simpleType>
```

SqlMessage.Class: This required element of simple type **nonNegativeInteger** describes the severity level of the server message. The range of values is defined by the server and subject to change, but it MUST be of type XML int. Refer to [MSDN-DEES] for details on the range of values defined by the server.

SqlMessage.LineNumber: This required element of simple type **nonNegativeInteger** describes the line number in the query that generated the server message. The value range is from 0 to 2147483647.

SqlMessage.Message: This required element of string type describes the text of the server message.

SqlMessage.Number: This required element of simple type **nonNegativeInteger** describes the identity number of the server message. The set of identity numbers is defined by the server.

SqlMessage.Procedure: This optional element of string type describes the name of the server-side method that generated this server message.

SqlMessage.Server: This optional element of string type describes the name of the server that generated this server message.

SqlMessage.Source: This required element of string type describes the name of the source that generated this server message.<1>

SqlMessage.State: This required element of simple type **nonNegativeInteger** describes the server state that generated the server message. Some messages apply to multiple server scenarios and this state number is used to identify the scenario that generated the message. The set of state numbers is defined by the server. The value range is from 0 to 2147483647.

~~2.1.1.1.3.1.4~~2.2.2.1.1.4 ~~2.2.2.1.1.4~~ **sqlbatchResult.SqlRowCount**

sqlbatchResponse.sqlbatchResult.SqlRowCount: This element of complex type **SqlRowCount** describes the portion of the response representing the affected rows resulting from the query. The affected rows include the number of rows in the resultset and the number of rows inserted/deleted/updated, and so on. The **SqlRowCount** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlRowCount" namespace as the following.

```
<xsd:complexType name="SqlRowCount">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="Count" type="xsd:long" />
  </xsd:sequence>
</xsd:complexType>
```

SqlRowCount.Count: This required element of XML long type describes the resulting number of rows affected by the query statement specified in the **BatchCommands** element of the request.

~~2.1.1.1.3.1.5~~2.2.2.1.1.5 ~~2.2.2.1.1.5~~ **sqlbatchResult.SqlResultCode**

sqlbatchResponse.sqlbatchResult.SqlResultCode: This element of simple type **SqlResultCode** describes the portion of the response that represents the return value of the entire request, if any. The **SqlResultCode** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types" namespace as the following.

```
<xsd:simpleType name="SqlResultCode">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="0" />
  </xsd:restriction>
</xsd:simpleType>
```

SqlResultCode: This simple type describes the return value of the entire request. The server may choose to specify a return value. The data is of type XML int and has a value range from 0 to 2147483647.

~~2.1.1.1.3.1.6~~2.2.2.1.1.6 ~~2.2.2.1.1.6~~ **sqlbatchResult.SqlTransaction**

sqlbatchResponse.sqlbatchResult.SqlTransaction: This element of complex type **SqlTransaction** describes the portion of the response representing the intermediary transaction token that can be used to associate a request to a particular transaction. The **SqlTransaction** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlTransaction" namespace as the following.

```
<xsd:complexType name="SqlTransaction">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="Descriptor" type="xsd:base64Binary" />
  </xsd:sequence>
</xsd:complexType>
```

```

<xsd:element name="Type">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Begin"/>
      <xsd:enumeration value="Commit"/>
      <xsd:enumeration value="Rollback"/>
      <xsd:enumeration value="EnlistDTC"/>
      <xsd:enumeration value="Defect"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

```

SqlTransaction.Descriptor: This required element of XML base64 binary type describes the token value of the transaction ID.

SqlTransaction.Type: This required element of simple enumeration type describes the state of the transaction. The supported values are listed in the following table.

Value	Meaning
Begin	The transaction has been started. The token value specified within the Descriptor element MUST be used to rejoin/commit/roll back/defect the transaction.
Commit	The transaction has been committed. The token value specified within the Descriptor element specifies the specific transaction that was committed.
Rollback	The transaction has been rolled back. The token value specified within the Descriptor element specifies the specific transaction that was rolled back.
EnlistDTC	The distributed transaction has been started. The token value specified within the Descriptor element MUST be used to rejoin/commit/roll back/defect the transaction.
Defect	The distributed transaction has been stopped. The token value specified within the Descriptor element specifies the specific transaction that was stopped.

~~2.1-1.1-42.2.2.2.2 2.2-2.2.2~~ sqlbatchSoapOut SOAP Header

Besides the required SOAP body, the **sqlbatchSoapOut** message also supports the optional SOAP header, **sqlSession**. The SOAP header that is allowed is defined in the "http://schemas.microsoft.com/sqlserver/2004/SOAP/Options" namespace.

```

<xsd:element name="sqlSession">
  <xsd:annotation>
    <xsd:documentation>SQL Server SOAP
    Session</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="terminate"
    default="false" type="xsd:boolean" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>Set to 'true' to request to
        terminate an existing session.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="sessionId" type="xsd:base64Binary"
    form="unqualified">
      <xsd:annotation>
        <xsd:documentation>The ID of a

```

```

session.</xsd:documentation>
  </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="timeout" type="xsd:int"
form="unqualified">
  <xsd:annotation>
  <xsd:documentation>The timeout in seconds before
the session expires.</xsd:documentation>
  </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="transactionDescriptor"
type="xsd:base64Binary" form="unqualified">
  <xsd:annotation>
  <xsd:documentation>The descriptor of a
transaction to enlist to.</xsd:documentation>
  </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
</xsd:element>

```

The SOAP header element does not make use of the standard SOAP `mustUnderstand` attribute and the `actor` attribute. The details of the SOAP header are described in section 2.2.2.2.1.

~~2.1.1.1.4.12.2.2.2.1~~ ~~2.2.2.2.1~~ **sqlSession SOAP Header**

sqlSession: This optional element provides the ability for the server to notify the client about the current state of a named query session initiated by the client. The current state is provided to the client through the use of the associated optional attributes described in the following paragraphs.

sqlSession.terminate: This optional attribute of Boolean type describes whether or not the server has terminated the session specified by the **sessionId** attribute. This attribute has a default value of false. The allowed values are listed in the following table.

Value	Meaning
true	The server terminated the session specified by the sessionId attribute.
false	The server did not terminate the session specified by the sessionId attribute.

sqlSession.sessionId: This optional attribute of XML base64 binary type specifies the session token. This token is first generated by the server when a new session is created. If the client sent a request with a valid `sessionId`, then the server **MUST** specify the same `sessionId` in the response.

sqlSession.timeout: This optional attribute of XML int type specifies the duration, in seconds, of inactivity in the session before the server will terminate the session.

sqlSession.transactionDescriptor: This optional attribute of XML base64 binary type specifies the transaction token. This token is first returned by the server in the **sqlbatchResponse.sqlbatchResult.SqlTransaction** element of the `sqlbatchSoapOut` message, described in the **SqlTransaction.Descriptor**. If the client sent a request with a valid **transactionDescriptor**, then the server **SHOULD** specify the same **transactionDescriptor** in the response.

~~2.1.22.2.3~~ ~~2.2.3~~ **Elements**

None.

2.2.4—Complex Types

2.2.4 Complex Types

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

Complex type	Description
ArrayOfSqlParameter	Defines the list of parameters associated with the WSDL message.

2.2.4.1—ArrayOfSqlParameter

2.2.4.1 ArrayOfSqlParameter

This common complex type is used to specify the set of input/output parameters associated with the WSDL message. This protocol does not bound the upper limit of the number of occurrences of the **SqlParameter** element, but the upper application layer can determine a limit.

The following set of XML namespaces is used throughout this section:

xmlns:sqlparameter=

"http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"

xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"

Referenced by the **Parameters** element in both the sqlbatchSoapIn and the sqlbatchSoapOut WSDL message, the **ArrayOfSqlParameter** complex type is defined under the "http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter" namespace as the following.

```
<xsd:complexType name="ArrayOfSqlParameter">
  <xsd:sequence>
    <xsd:element minOccurs="0"
maxOccurs="unbounded" name="SqlParameter"
type="sqlparameter:SqlParameter" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SqlParameter">
  <xsd:sequence>
    <xsd:element minOccurs="1" maxOccurs="1"
name="Value" nillable="true" />
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string"
use="required" form="unqualified" />
  <xsd:attribute default="NVarChar" name="sqlDbType"
type="sqltypes:sqlDbTypeEnum" use="optional"
form="unqualified" />
  <xsd:attribute default="Input" name="direction"
type="sqlparameter:ParameterDirection" use="optional"
form="unqualified" />
  <xsd:attribute default="1" name="maxLength"
type="xsd:long" use="optional" form="unqualified" />
  <xsd:attribute default="18" name="precision"
type="xsd:unsignedByte" use="optional"
form="unqualified" />
  <xsd:attribute default="0" name="scale"
type="xsd:unsignedByte" use="optional"
form="unqualified" />
  <xsd:attribute default="" name="clrTypeName" />
</xsd:complexType>
```

```

type="xsd:string" use="optional" form="unqualified" />
  <xsd:attribute default="Default" name="sqlCompareOptions"
type="sqltypes:sqlCompareOptionsList" use="optional"
form="unqualified" />
  <xsd:attribute default="-1" name="localeId"
type="xsd:int" use="optional" form="unqualified" />
  <xsd:attribute default="0" name="sqlCollationVersion"
type="xsd:int" use="optional" form="unqualified" />
  <xsd:attribute default="0" name="sqlSortId"
type="xsd:int" use="optional" form="unqualified" />
  <xsd:attribute default="" name="xmlSchemaCollection"
type="xsd:string" use="optional" form="unqualified" />
</xsd:complexType>

<xsd:simpleType name="ParameterDirection">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Input" />
    <xsd:enumeration value="InputOutput" />
  </xsd:restriction>
</xsd:simpleType>

```

Details of the **ArrayOfSqlParameter** complex type are described in sections 2.2.4.1.1, 2.2.4.1.2, and 2.2.5.3.

~~2.1.2.1-12.2.4.1.1~~ ~~2.2.4.1.1~~ **SqlParameter**

SqlParameter: This complex type element defines the individual parameters that are associated with a query. When specified as part of the sqlbatchSoapIn WSDL message, this element represents an input parameter. The properties of the input parameter are defined by the various attributes and subelements that are associated with this element. When specified as part of the sqlbatchSoapOut WSDL message, this element represents an output parameter. The properties of the output parameter are defined by the various attributes and subelements that are associated with this element.

SqlParameter.name: This string type attribute MUST exist if a **SqlParameter** element is specified. This attribute is used to specify the name of the parameter. The value of this attribute is limited to 127 characters, which is one less than the maximum number of characters allowed in a SQL Server identifier; one character is reserved for the parameter ~~name's~~ implied "@'"@ character.

SqlParameter.sqlDbType: This enumeration simple type attribute may exist if a **SqlParameter** element is specified. This attribute is used to specify the SQL Server data type that the parameter value MUST be treated as by the server. The default value of this attribute is "NVarChar". The set of supported values is defined by the **sqlDbTypeEnum** simple type, which is documented in section 2.2.5.3.

SqlParameter.direction: This enumeration simple type attribute may exist if a **SqlParameter** element is specified. This attribute is used to specify the direction of the parameter. The default value of this attribute is "Input". The supported values are listed in the following table.

Value	Meaning
Input	This parameter is an input-only parameter. The server will not return any values for this parameter.
InputOutput	This parameter is an input and output parameter. The server will return a value for this parameter.

SqlParameter.maxLength: This long type attribute may exist if a **SqlParameter** element is specified. The default value of this attribute is "1". This attribute is used to specify the maximum length of the parameter defined by the following **sqlDbType** data types.

sqlDbType	Value range
Binary	0 – 8000
VarBinary	-1, 0 – 8000 Note: -1 denotes varbinary (max)
Char	0 – 8000
NChar	0 – 4000
NVarChar	-1, 0 – 4000 Note: -1 denotes nvarchar (max)
VarChar	-1, 0 – 8000 Note: -1 denotes varchar (max)

SqlParameter.precision: This unsigned byte type attribute may exist if a **SqlParameter** element is specified. The default value of this attribute is "18". This attribute is used to specify the precision of the parameter defined by the following **sqlDbType** data types.

sqlDbType	Value range
Decimal	0 – 38
Float	0 – 38
Money	0 – 3
Real	0 – 38
SmallMoney	0 – 3

SqlParameter.scale: This unsigned byte type attribute may exist if a **SqlParameter** element is specified. The default value of this attribute is "0". This attribute is used to specify the scale of the parameter that is defined by the following **sqlDbType** data types.

sqlDbType	Value range
Decimal	0 – 38
Float	0 – 38
Real	0 – 38

SqlParameter.clrTypeName: This string type attribute may exist if a **SqlParameter** element is specified. This attribute is used to specify the name of the common language runtime (CLR) data type for the parameter when the **sqlDbType** attribute has a value of "Udt". The default value of this attribute is an empty string (""). The set of supported values depends on the set of CLR user-defined type (UDT) values defined in the SQL Server instance. The full three-part name of the CLR UDT SHOULD be used when specifying this attribute value.

SqlParameter.sqlCompareOptions: This enumeration simple type attribute may exist if a **SqlParameter** element is specified. This attribute is used by SQL Server string data types to specify how character values are compared and sorted. The default value of this attribute is "Default", which defers to the setting defined by the connected server. The set of supported values is defined by the **sqlCompareOptionsList** simple type, described in section 2.2.5.1.

SqlParameter.localeId: This int type attribute may exist if a **SqlParameter** element is specified. This attribute is used to specify the collation of the parameter character value. The default value

of this attribute is "-1". A value of "-1" tells the server to use the locale of the current database. For example, if the parameter is of data type varchar and the locale is specified as Japanese, the server converts the ~~parameter's~~ XML character value to Japanese.

Note Specifying a localeId value that is different from the current database localeId may cause additional data conversions, depending on the query.

SqlParameter.sqlCollationVersion: This int type attribute may exist if a **SqlParameter** element is specified. This attribute is used by SQL Server string data types to specify the version of the collation. The default value of this attribute is "0".<2>

SqlParameter.sqlSortId: This int type attribute may exist if a **SqlParameter** element is specified. This attribute is used by SQL Server string data types to specify the SQL Server sort id. The default value of this attribute is "0". The set of supported values is defined by [MSDN-SQLCollation].

SqlParameter.xmlSchemaCollation: This string type attribute may exist if a **SqlParameter** element is specified. This attribute is used to specify the name of the XML schema collection of the parameter when the **sqlDbType** attribute has a value of "Xml". The default value of this attribute is empty string (""). The set of supported values depends on the set of XML schema collections defined in the SQL Server instance. The full three-part name of the XML schema collection SHOULD be used when specifying this attribute value.

~~2.1.2.1-2.2.4.1.2~~ ~~2.2.4.1.2~~ **SqlParameter.Value**

SqlParameter.Value: This required element MUST exist if a **SqlParameter** element is specified. This element is used to specify the value of the parameter. All the standard rules of XML element values apply. To specify a binary value, it MUST be encoded in base64 encoding. To specify XML data values and CLR UDT data values, the value is specified in text XML format [MSDN-XMLSNET] or a serialization format defined by the implementer of the CLR UDT data type. All other data type values are specified in standard XML string value format.

Following the standard [XML10/5] rules, this element supports specifying the standard xsi:type attribute. The list of supported values for the xsi:type attribute is limited to standard XML data types and the types defined under the following "http://schemas.microsoft.com/sqlserver/2004/sqltypes" namespace. Refer to section 2.2.5.2 for the complete list. If an unknown xsi:type value is specified, the server returns a **SOAP fault** message containing "UnsupportedNamespaceInXsiTypeAttribute".

~~2.2.5~~ **Simple Types**

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
sqlCompareOptionsList	Defines the set of options used for string value comparisons associated with a string parameter or result data column.
sqlTypes	Defines the set of data types that represent supported SQL Server data types in terms of the XML data type.

2.2.5.1 sqlCompareOptionsList

2.2.5.1 sqlCompareOptionsList

This common simple type is used to specify the set of string value comparison options that are associated with a string parameter or result data column. The following XML namespace is used throughout this section:

xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"

Referenced by the **sqlCompareOptions** attribute within the **ArrayOfSqlParameter** complex type, the **sqlCompareOptionsList** type is defined under the "http://schemas.microsoft.com/sqlserver/2004/sqltypes" namespace as the following.

```
<xsd:simpleType name="sqlCompareOptionsList">
  <xsd:list itemType="sqltypes:sqlCompareOptionsEnum" />
</xsd:simpleType>

<xsd:simpleType name="sqlCompareOptionsEnum">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Default" />
    <xsd:enumeration value="None" />
    <xsd:enumeration value="IgnoreCase" />
    <xsd:enumeration value="IgnoreNonSpace" />
    <xsd:enumeration value="IgnoreKanaType" />
    <xsd:enumeration value="IgnoreWidth" />
    <xsd:enumeration value="BinarySort" />
    <xsd:enumeration value="BinarySort2" />
  </xsd:restriction>
</xsd:simpleType>
```

The **sqlCompareOptionsList** simple type is a list of values that are defined by the **sqlCompareOptionsEnum** simple enumeration type. Each compare option tells the server how string comparisons should be evaluated when using Windows collations. The supported values are listed in the following table.<3>

Value	Meaning
Default	Use the database default string comparison settings.
IgnoreCase IgnoreNonSpace IgnoreKanaType IgnoreWidth	Case-insensitive, accent-insensitive, kana-insensitive, width-insensitive.
IgnoreCase IgnoreNonSpace IgnoreWidth	Case-insensitive, accent-insensitive, kana-sensitive, width-insensitive.
IgnoreCase IgnoreNonSpace	Case-insensitive, accent-insensitive, kana-sensitive, width-sensitive.
IgnoreCase IgnoreNonSpace IgnoreKanaType	Case-insensitive, accent-insensitive, kana-insensitive, width-sensitive.
IgnoreCase IgnoreKanaType IgnoreWidth	Case-insensitive, accent-sensitive, kana-insensitive, width-insensitive.
IgnoreCase IgnoreWidth	Case-insensitive, accent-sensitive, kana-sensitive, width-insensitive.
IgnoreCase	Case-insensitive, accent-sensitive, kana-sensitive, width-sensitive.

Value	Meaning
IgnoreCase IgnoreKanaType	Case-insensitive, accent-sensitive, kana-insensitive, width-sensitive.
IgnoreNonSpace IgnoreKanaType IgnoreWidth	Case-sensitive, accent-insensitive, kana-insensitive, width-insensitive.
IgnoreNonSpace IgnoreWidth	Case-sensitive, accent-insensitive, kana-sensitive, width-insensitive.
IgnoreNonSpace	Case-sensitive, accent-insensitive, kana-sensitive, width-sensitive.
IgnoreNonSpace IgnoreKanaType	Case-sensitive, accent-insensitive, kana-insensitive, width-sensitive.
IgnoreKanaType IgnoreWidth	Case-sensitive, accent-sensitive, kana-insensitive, width-insensitive.
IgnoreWidth	Case-sensitive, accent-sensitive, kana-sensitive, width-insensitive.
IgnoreKanaType	Case-sensitive, accent-sensitive, kana-insensitive, width-sensitive.
None	Case-sensitive, accent-sensitive, kana-sensitive, width-sensitive.

~~2.2.5.2 sqlTypes~~

2.2.5.2 sqlTypes

The following set of simple types defines SQL Server data types in terms of XML data types. These types are defined under the "http://schemas.microsoft.com/sqlserver/2004/sqltypes" namespace, typically referred to using the sqltypes prefix.

```

<xsd:simpleType name="char">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="nchar">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="varchar">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="nvarchar">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="text">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="ntext">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="varbinary">
  <xsd:restriction base="xsd:base64Binary"/>
</xsd:simpleType>
<xsd:simpleType name="binary">
  <xsd:restriction base="xsd:base64Binary"/>
</xsd:simpleType>
<xsd:simpleType name="image">
  <xsd:restriction base="xsd:base64Binary"/>
</xsd:simpleType>

```

```

<xsd:simpleType name="timestamp">
  <xsd:restriction base="xsd:base64Binary">
    <xsd:maxLength value="8"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="timestampNumeric">
  <!-- The timestampNumeric type supports a
  legacy format of timestamp. -->
  <xsd:restriction base="xsd:long"/>
</xsd:simpleType>
<xsd:simpleType name="decimal">
  <xsd:restriction base="xsd:decimal"/>
</xsd:simpleType>
<xsd:simpleType name="numeric">
  <xsd:restriction base="xsd:decimal"/>
</xsd:simpleType>
<xsd:simpleType name="bigint">
  <xsd:restriction base="xsd:long" />
</xsd:simpleType>
<xsd:simpleType name="int">
  <xsd:restriction base="xsd:int"/>
</xsd:simpleType>
<xsd:simpleType name="smallint">
  <xsd:restriction base="xsd:short"/>
</xsd:simpleType>
<xsd:simpleType name="tinyint">
  <xsd:restriction base="xsd:unsignedByte"/>
</xsd:simpleType>
<xsd:simpleType name="bit">
  <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>
<xsd:simpleType name="float">
  <xsd:restriction base="xsd:double"/>
</xsd:simpleType>
<xsd:simpleType name="real">
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="datetime">
  <xsd:restriction base="xsd:dateTime">
    <xsd:pattern value="((000[1-9])|(00[1-9][0-9])|(0[1-9][0-9]{2})|([1-9][0-9]{3}))-(00[1-9])|(1[012]))-(00[1-9])|([12][0-9])|(3[01]))T((01[0-9])|(2[0-3]))(:[0-5][0-9])?(\.[0-9]{2}[037])?"/>
    <xsd:maxInclusive value="9999-12-31T23:59:59.997"/>
    <xsd:minInclusive value="1753-01-01T00:00:00.000"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="smalldatetime">
  <xsd:restriction base="xsd:dateTime">
    <xsd:pattern value="((000[1-9])|(00[1-9][0-9])|(0[1-9][0-9]{2})|([1-9][0-9]{3}))-(00[1-9])|(1[012]))-(00[1-9])|([12][0-9])|(3[01]))T((01[0-9])|(2[0-3]))(:[0-5][0-9])(:00)"/>
    <xsd:maxInclusive value="2079-06-06T23:59:00"/>
    <xsd:minInclusive value="1900-01-01T00:00:00"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="money">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits value="19"/>
    <xsd:fractionDigits value="4"/>
    <xsd:maxInclusive value="922337203685477.5807"/>
    <xsd:minInclusive value="-922337203685477.5808"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="smallmoney">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits value="10"/>
    <xsd:fractionDigits value="4"/>
  </xsd:restriction>

```

```

        <xsd:maxInclusive value="214748.3647"/>
        <xsd:minInclusive value="-214748.3648"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="uniqueidentifier">
    <xsd:restriction base="xsd:string">
        <xsd: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})|(\{[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}\})"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- sql_variant directly maps to xsd:anyType -->
<xsd:complexType name="xml" mixed="true">
    <xsd:sequence>
        <xsd:any minOccurs="0" maxOccurs="unbounded"
processContents="skip" />
    </xsd:sequence>
</xsd:complexType>
<!-- the following type is for FOR XML binary URL
results only -->
<xsd:simpleType name="dbobject">
    <xsd:restriction base="xsd:anyURI" />
</xsd:simpleType>

```

The SQL Server equivalent data types are listed in the following table.

Value	Meaning
sqltypes:char	SQL Server char data type.
sqltypes:nchar	SQL Server nchar data type.
sqltypes:varchar	SQL Server varchar data type.
sqltypes:nvarchar	SQL Server nvarchar data type.
sqltypes:text	SQL Server text data type.
sqltypes:ntext	SQL Server ntext data type.
sqltypes:varbinary	SQL Server varbinary data type.
sqltypes:binary	SQL Server binary data type.
sqltypes:image	SQL Server image data type.
sqltypes:timestamp	SQL Server timestamp data type.
sqltypes:decimal	SQL Server decimal data type.
sqltypes:numeric	SQL Server numeric data type.
sqltypes:bigint	SQL Server bigint data type.
sqltypes:int	SQL Server int data type.
sqltypes:smallint	SQL Server smallint data type.
sqltypes:tinyint	SQL Server tinyint data type.
sqltypes:bit	SQL Server bit data type.

Value	Meaning
sqltypes:float	SQL Server float data type.
sqltypes:real	SQL Server real data type.
sqltypes:datetime	SQL Server datetime data type.
sqltypes:smalldatetime	SQL Server smalldatetime data type.
sqltypes:money	SQL Server money data type.
sqltypes:smallmoney	SQL Server smallmoney data type.
sqltypes:uniqueidentifier	SQL Server uniqueidentifier data type.
sqltypes:xml	SQL Server xml data type.
xsd:any	SQL Server CLR UDT data type.

In this table, the CLR UDT data type is mapped to XML schema of `xsd:any`. This protocol itself does not define the XML structure of a user-defined type value; it only defines the mechanism by which a user-defined type value can be exchanged. It is the user-defined type ~~creator's~~creator's responsibility to define the XML representation for the type and to provide the XML schema of the type to the client application. The transfer of the XML schema between the user-defined type creator and the client application is done out of band. For additional information about using CLR's XML serialization attributes to control the XML serialization format, refer to [MSDN-XMLSNET].

2.2.5.3 ~~sqlDbTypeEnum~~

2.2.5.3 sqlDbTypeEnum

This **sqlDbTypeEnum** simple type is defined under the "http://schemas.microsoft.com/sqlserver/2004/sqltypes" namespace as the following.

```
<xsd:simpleType name="sqlDbTypeEnum">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="BigInt" />
    <xsd:enumeration value="Binary" />
    <xsd:enumeration value="Bit" />
    <xsd:enumeration value="Char" />
    <xsd:enumeration value="DateTime" />
    <xsd:enumeration value="Decimal" />
    <xsd:enumeration value="Float" />
    <xsd:enumeration value="Image" />
    <xsd:enumeration value="Int" />
    <xsd:enumeration value="Money" />
    <xsd:enumeration value="NChar" />
    <xsd:enumeration value="NText" />
    <!-- The sqlDbTypeEnum aligns with
the .Net System.Data.SqlDbType enum and
does not provide an entry for
Numeric (which is mapped to Decimal). -->
    <xsd:enumeration value="NVarChar" />
    <xsd:enumeration value="Real" />
    <xsd:enumeration value="SmallDateTime" />
    <xsd:enumeration value="SmallInt" />
    <xsd:enumeration value="SmallMoney" />
    <xsd:enumeration value="Text" />
    <xsd:enumeration value="Timestamp" />
    <xsd:enumeration value="TinyInt" />
    <xsd:enumeration value="Udt" />
  </xsd:restriction>
</xsd:simpleType>
```

```

    <xsd:enumeration value="UniqueIdentifier" />
    <xsd:enumeration value="VarBinary" />
    <xsd:enumeration value="VarChar" />
    <xsd:enumeration value="Variant" />
    <xsd:enumeration value="Xml" />
  </xsd:restriction>
</xsd:simpleType>

```

The **sqlDbTypeEnum** simple type defines the set of supported values for the **sqlDbType** attribute of the **SqlParameter** element. The equivalent SQL Server data types for each of the **sqlDbTypeEnum** values are listed in the following table.

Value	Meaning
BigInt	Bigint
Binary	Binary
Bit	Bit
Char	Char
DateTime	Datetime
Decimal	Decimal; numeric
Float	Float
Image	Image
Int	Int
Money	Money
NChar	Nchar
NText	Ntext
NVarChar	Nvarchar
Real	Real
SmallDateTime	Smalldatetime
SmallInt	Smallint
SmallMoney	Smallmoney
Text	Text
Timestamp	Timestamp
TinyInt	Tinyint
Udt	UDT; CLR UDT
UniqueIdentifier	globally unique identifier (GUID) ; UniqueID
VarBinary	Varbinary
VarChar	Varchar
Variant	Sql_variant

Value	Meaning
Xml	XML; schema-bounded XML

~~2.1.3~~~~2.2.6~~ ~~2.2.6~~ **Attributes**

None.

~~2.1.4~~~~2.2.7~~ ~~2.2.7~~ **Groups**

None.

~~2.1.5~~~~2.2.8~~ ~~2.2.8~~ **Attribute Groups**

None.

~~2.2.3~~~~2.3~~ ~~2.3~~ **Directory Service Schema Elements**

None.

3 ~~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.

3.1 ~~3.1~~—Batch_EPSoap Server Details

This section describes the server behavior of the NWS protocol. This port type supports one WSDL operation: sqlbatch.

~~3.1.1~~—Abstract Data Model

3.1.1 Abstract Data Model

This section describes a conceptual model of the possible data organization that an implementation maintains to participate in this protocol. This information helps to explain how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document.

Note The Native Web Services (NWS) protocol is request/response based. Each request/response pair can be encapsulated within its own NWS session (which is the default behavior) or within a named session that is initiated by the client.

3.1.1.1 ~~3.1.1.1~~—Session-specific Structures

The following structure is required per session to implement session management:

- ~~▪~~ **NWS.sessionId**: An optional unique identifier for the session. This variable is used to allow the client to reuse a fully logged-in session beyond a single request.
- ~~▪~~ **NWS.timeout**: The length of time for which the client requests that a reusable session be stored before it is removed from the pool of sessions.

For more information, see section 3.1.2.

Note The dotted notation indicates the structure of an instance of an **NWS object**. For example, **NWS.sessionId** refers to the **sessionId** variable of the NWS object.

All data that is sent in response to a client request (for example, **RowSet** element data, metadata and XML data) is not maintained as part of the session state; rather, the data is maintained by the higher layer of the server, which uses the NWS protocol to return its data to the client.

~~3.1.2~~—Timers

3.1.2 Timers

The NWS protocol has an authentication timer and a session timer that the server SHOULD implement. The protocol does not have a timer on a data stream.

~~3.1.23.1.3~~ ~~3.1.3~~—Initialization

The server MUST establish a listening endpoint based on the transport protocols described in section 2.1.

The server may establish additional listening endpoints.<4>

When a client makes a connection request, the transport layer listening endpoint SHOULD initialize all resources required for this connection so that the server will be ready to receive a sqlbatchSoapIn message.

~~3.1.3~~3.1.4 ~~3.1.4~~ Message Processing Events and Sequencing Rules

Operation	Description
sqlbatch	This operation enables the client application to execute ad-hoc SQL Server query statements, including parameterized statements.

~~3.1.4.1~~ Single sqlbatch

3.1.4.1 Single sqlbatch

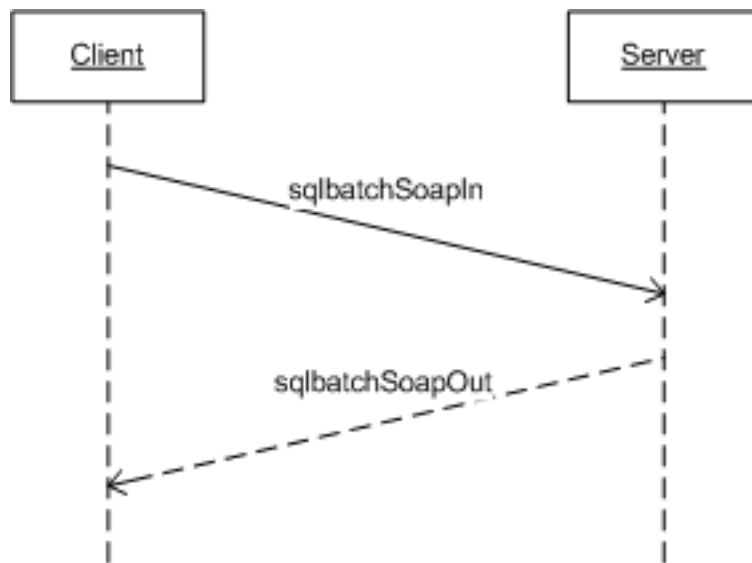


Figure 44: Single sqlbatch operation

This section and its subsections describe the scenario in which the client sends one-off sqlbatch requests to the server. The sqlbatch operation has an input message named **sqlbatchSoapIn** and an output message named **sqlbatchSoapOut**, as shown in the following WSDL snippet for this operation.

```
<wsdl:operation name="sqlbatch">
  <wsdl:input message="tns:sqlbatchSoapIn"/>
  <wsdl:output message="tns:sqlbatchSoapOut"/>
</wsdl:operation>
```

As shown in this figure, only one SOAP message sequence is required to execute a SQL Server query statement. At any time throughout the entire SOAP message parsing process, if there are any parsing errors, the server returns a SOAP fault message and closes the connection.

~~3.1.3~~3.1.4.1.1 ~~3.1.4.1.1~~ Messages

The following WSDL message definitions are specific to this operation.

~~3.1.3.1.1.13.1.4.1.1.1~~ ~~3.1.4.1.1.1~~ **sqlbatchSoapIn**

```
<wsdl:message name="sqlbatchSoapIn">
  <wsdl:part name="parameters" element="sql:sqlbatch"/>
</wsdl:message>
```

~~3.1.3.1.1.23.1.4.1.1.2~~ ~~3.1.4.1.1.2~~ **sqlbatchSoapOut**

```
<wsdl:message name="sqlbatchSoapOut">
  <wsdl:part name="parameters" element="sql:sqlbatchResponse"/>
</wsdl:message>
```

~~3.1.3.1.23.1.4.1.2~~ ~~3.1.4.1.2~~ **Elements**

The following XML schema element definitions are specific to this operation.

~~3.1.3.1.2.13.1.4.1.2.1~~ ~~3.1.4.1.2.1~~ **sqlbatch**

This element structure is described by the **sqlbatchSoapIn** WSDL message in section 2.2.2.1.

The server parses through the SOAP body to look for the **sqlbatch** element. Once the **sqlbatch** element is found, the server parses the **BatchCommands** element and the **Parameters** element. It then passes the data to the upper layer for execution. As mentioned earlier, if there are any parsing errors, the server MUST return a SOAP fault message. If an error occurs during the execution in the upper layer, the upper layer SHOULD return the error in a **SqlMessage** element as part of the **SqlResultStream**. The upper layer may determine that it cannot return the error in a **SqlMessage** element and terminate the connection.

~~3.1.3.1.2.23.1.4.1.2.2~~ ~~3.1.4.1.2.2~~ **sqlbatchResponse**

This element structure is described by the **sqlbatchSoapOut** WSDL message in section 2.2.2.2.

When the upper layer finishes execution, the upper layer provides the data to be sent back to the client. The server sends the SOAP message response using the **sqlbatchResponse** element. The result of the query execution is returned in the **sqlbatchResult** element, in accordance with the **SqlResultStream** complex type, while output parameters, if any, are returned in the **Parameters** element, in accordance with the **ArrayOfSqlParameter** complex type.

~~3.1.3.1.33.1.4.1.3~~ ~~3.1.4.1.3~~ **Complex Types**

The XML schema complex type definitions specific to this operation are described in sections 3.1.4.1.3.1 and 3.1.4.1.3.2.

~~3.1.3.1.3.13.1.4.1.3.1~~ ~~3.1.4.1.3.1~~ **SqlResultStream**

When sending a response, the server generates the XML as defined by this complex type as part of the **sqlbatchResponse** element to return the query execution result.

~~3.1.3.1.3.23.1.4.1.3.2~~ ~~3.1.4.1.3.2~~ **ArrayOfSqlParameter**

When the server encounters this complex type as part of the **sqlbatch** element, the server creates and defines the list of parameters specified by this complex type.

When sending a response, the server generates the XML as defined by this complex type as part of the **sqlbatchResponse** element to return output parameter values.

~~3.1.3.1.43.1.4.1.4~~ ~~3.1.4.1.4~~ **Simple Types**

The XML schema simple type definitions specific to this operation are described in section 3.1.4.1.4.1.

~~3.1.3.1.4.13.1.4.1.4.1~~ ~~3.1.4.1.4.1~~ **ParameterDirection**

When the server is processing each **SqlParameter** element, it may encounter this simple type, which defines whether the parameter is an input-only parameter or an input/output parameter. Refer to the **SqlParameter.direction** element in section 2.2.4.1.1 for details.

~~3.1.3.1.53.1.4.1.5~~ ~~3.1.4.1.5~~ **Attributes**

None.

~~3.1.3.1.63.1.4.1.6~~ ~~3.1.4.1.6~~ **Groups**

~~None.~~

~~3.1.4.1.7~~ **Attribute Groups**

None.

~~3.1.4.2~~ **Session-based sqlbatch**

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 Session-based sqlbatch

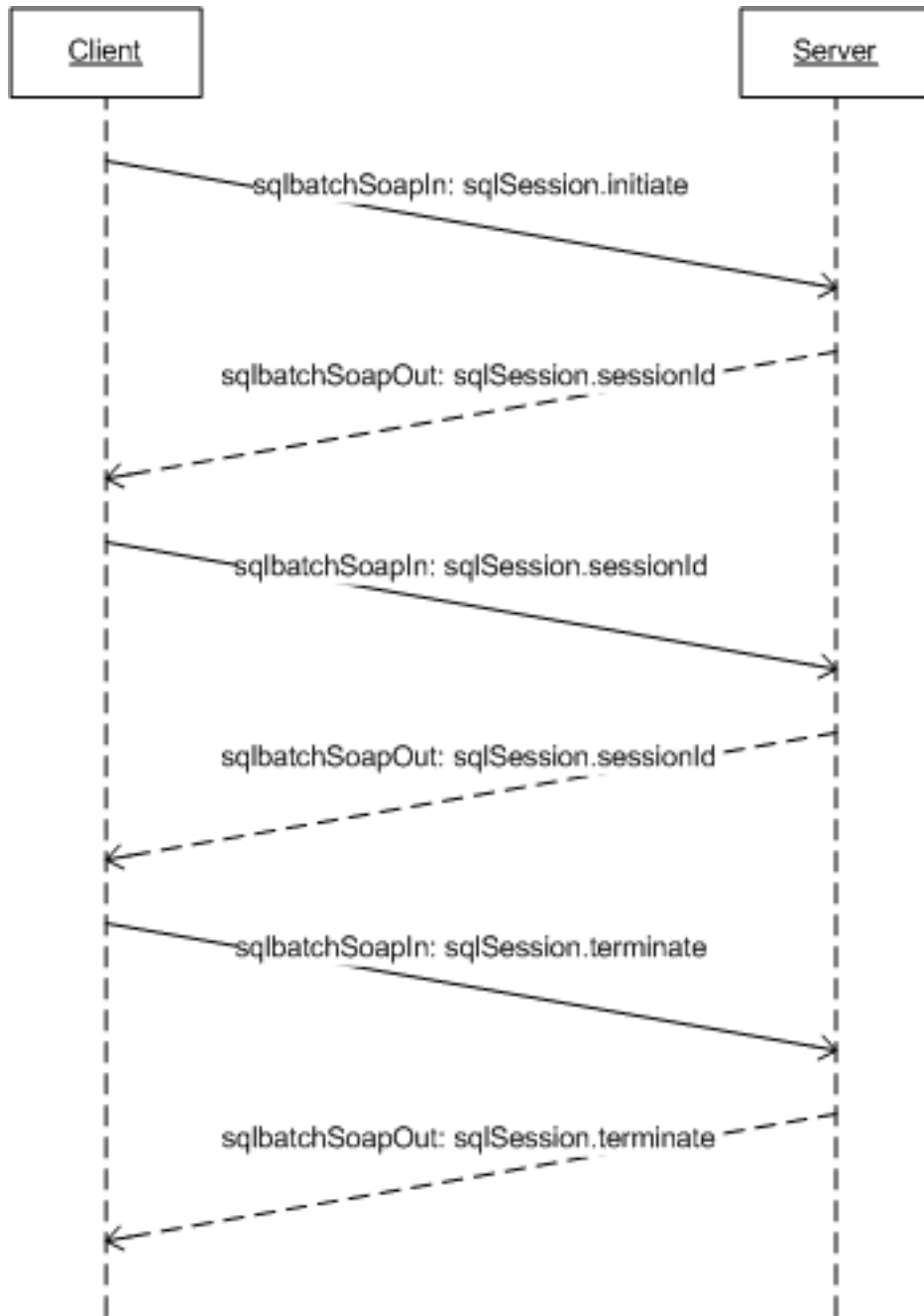


Figure 55: Session-based sqlbatch operation

This section and its subsections describe the scenario in which the client is establishing a session to enable subsequent SOAP messages to participate in the same SQL Server session environment. The sqlbatch operation has an input message named **sqlbatchSoapIn** and an output message named **sqlbatchSoapOut**, as shown in the following WSDL snippet for this operation.

```
<wsdl:operation name="sqlbatch">
  <wsdl:input message="tns:sqlbatchSoapIn"/>
```

```
<wsdl:output message="tns:sqlbatchSoapOut"/>
</wsdl:operation>
```

As shown in this figure, the client MUST first send a **sqlbatchSoapIn** message with the **sqlSession** SOAP header, described in section 2.2.2.1.2.10, **Initiate** attribute set to true. The server then creates a NWS object, setting the NWS.sessionId value to the **sqlSession.sessionId** attribute (if specified), and setting the timeout value to the **sqlSession.timeout** attribute. In the **sqlbatchSoapOut** response message, if no errors occur, the server MUST set the **sqlSession** SOAP header, described in section 2.2.2.2.1, **sessionId** attribute to NWS.sessionId. The server SHOULD also set the **sqlSession.timeout** attribute to NWS.timeout.

The client can then send additional **sqlbatchSoapIn** messages with the same ID value for the **sqlSession.sessionId** attribute so that each message is executed in the same server environment context of the existing NWS object where the NWS.sessionId value is equal to the **sqlSession.sessionId** attribute. When the client is done with the session, the client SHOULD send a **sqlbatchSoapIn** message with the **sqlSession.sessionId** attribute set to the ID value and the **sqlSession.terminate** attribute set to true. This allows the server to release any resources held by the session, such as the associated NWS object, without the need to wait until the timeout expires. If no errors occur, the server MUST send a **sqlbatchSoapOut** response message with the **sqlSession.terminate** attribute set to true and the **sqlSession.sessionId** attribute set to the value of the ID.

At any time throughout the various SOAP message exchanges, if there are any parsing errors, then the server returns a SOAP fault message and closes the connection.

~~3.1.3.1.73.1.4.2.1~~ ~~3.1.4.2.1~~ Messages

The following WSDL message definitions are specific to this operation.

~~3.1.3.1.7.13.1.4.2.1.1~~ ~~3.1.4.2.1.1~~ sqlbatchSoapIn

```
<wsdl:message name="sqlbatchSoapIn">
  <wsdl:part name="parameters" element="sql:sqlbatch"/>
</wsdl:message>
```

~~3.1.3.1.7.23.1.4.2.1.2~~ ~~3.1.4.2.1.2~~ sqlbatchSoapOut

```
<wsdl:message name="sqlbatchSoapOut">
  <wsdl:part name="parameters" element="sql:sqlbatchResponse"/>
</wsdl:message>
```

~~3.1.3.1.83.1.4.2.2~~ ~~3.1.4.2.2~~ Elements

The following XML schema element definitions are specific to this operation.

~~3.1.3.1.8.13.1.4.2.2.1~~ ~~3.1.4.2.2.1~~ sqlSession

This element structure is described in section 2.2.2.1.2.10.

When the server receives a **sqlbatchSoapIn** message, it parses the SOAP header to look for the **sqlSession** element. When the **sqlSession** element is found, the server parses the associated attributes. It then passes the data to the upper layer for execution. As mentioned earlier, if there are any parsing errors, the server MUST return a SOAP fault message. If an error occurs during the execution in the upper layer, the upper layer SHOULD return the error in a **SqlMessage** element as part of the **SqlResultStream** whenever possible.

When the server is returning a **sqlbatchSoapOut** message, the server-side upper layer determines whether a **sqlSession** element is to be specified as part of the response.

~~3.1.3.1.8.23.1.4.2.2.2~~ ~~3.1.4.2.2.2~~ **sqlbatch**

This element structure is defined by the WSDL message **sqlbatchSoapIn** as described in section 2.2.2.1.

The server parses through the SOAP body to look for the **sqlbatch** element. When the **sqlbatch** element is found, the server parses the **BatchCommands** element and the **Parameters** element. It then passes the data to the upper layer for execution. As mentioned above, if there are any parsing errors, the server MUST return a SOAP fault message. If an error occurs during the execution in the upper layer, the upper layer SHOULD return the error in a **SqlMessage** element as part of the **SqlResultStream** whenever possible. The upper layer may determine that it cannot return the error in a **SqlMessage** element and terminate the connection.

~~3.1.3.1.8.33.1.4.2.2.3~~ ~~3.1.4.2.2.3~~ **sqlbatchResponse**

This element structure is defined by the WSDL message **sqlbatchSoapOut** as described in section 2.2.2.2.

When the upper layer finishes execution, the upper layer provides the data to be sent back to the client. The server sends the SOAP message response by using the **sqlbatchResponse** element. The result of the query execution is returned in the **sqlbatchResult** element, in accordance with the **SqlResultStream** complex type, while output parameters, if any, are returned in the **Parameters** element, in accordance with the **ArrayOfSqlParameter** complex type.

~~3.1.3.1.93.1.4.2.3~~ ~~3.1.4.2.3~~ **Complex Types**

The following XML schema complex type definitions are specific to this operation.

~~3.1.3.1.9.13.1.4.2.3.1~~ ~~3.1.4.2.3.1~~ **SqlResultStream**

When sending a response, the server generates the XML as defined by this complex type as part of the **sqlbatchResponse** element to return the query execution result.

~~3.1.3.1.9.23.1.4.2.3.2~~ ~~3.1.4.2.3.2~~ **ArrayOfSqlParameter**

In the SOAP Body Parse state, when the server encounters this complex type, the server defines the list of parameters as specified by this complex type.

~~3.1.3.1.103.1.4.2.4~~ ~~3.1.4.2.4~~ **Simple Types**

The following XML schema simple type definitions are specific to this operation.

~~3.1.3.1.10.13.1.4.2.4.1~~ ~~3.1.4.2.4.1~~ **ParameterDirection**

When the server is processing each **SqlParameter** element, it may encounter this simple type. It defines whether the parameter is an input-only parameter or an input/output parameter. Refer to the **SqlParameter.direction** element in section 2.2.4.1.1 for details.

~~3.1.3.1.113.1.4.2.5~~ ~~3.1.4.2.5~~ **Attributes**

None.

~~3.1.3.1.123.1.4.2.6~~ ~~3.1.4.2.6~~ **Groups**

~~None.~~

~~3.1.4.2.7 Attribute Groups~~

None.

~~3.1.4.3 Authentication~~

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 Authentication

For the sequence of SOAP messages defined by this protocol, each SOAP message sent from the client to the server MUST have a valid authentication token in the HTTP header in the format defined by [RFC2616]. Any message exchanges that are necessary to validate the authentication token are in accordance with [RFC4559] and [RFC2617]. This protocol does not extend or enhance these RFC specifications.

This protocol does limit the use of Basic authentication, as defined by [RFC2617], to connections established using HTTPS.

In addition to specifying the authentication token in the HTTP headers, the client can also use the [WSSUTP] SOAP header to specify the name and password of a SQL Server user account. If the token is found, the server validates the user credentials that are specified in the [WSSUTP] header. If the token is not found, the server uses the user credentials specified in the HTTP Authorization header to log in to SQL Server.

~~3.1.5 Timer Events~~

3.1.5 Timer Events

As mentioned in section 3.1.2, the server SHOULD implement an authentication timer. This timer SHOULD have a timeout value of 30 seconds. This timer is triggered to start when the initial client SOAP message request with a valid HTTP authentication BLOB is received by the server. The timer is stopped and reset when the HTTP authentication meets the requirements as specified by [RFC2616]. If the timer starts before the HTTP authentication requirements are met, then the server will not return a response to the client and will close the connection.

The server SHOULD also implement a session timer. This timer SHOULD have a default timeout value of 60 seconds, but it can be configured to allow for a longer default timeout value. The client application can choose to request a different timeout value. However, the shorter timeout value, from either the system default or the request by the client application, MUST be used as the timeout value for this timer. This timer is triggered to start every time the server sends a session-based response to the client. The timer is stopped and reset when the client sends another session-based request to the same session. When the timeout expires, the server MUST terminate the session and close the connection.

~~3.1.4.3.1.6 3.1.6 Other Local Events~~

When there is a failure in under-layers, such as the network, the server SHOULD terminate the HTTP connection without sending any response to the client. An under-layer failure may be triggered by a network failure. It may also be triggered by the termination action from the client, which could be communicated to the server stack by the under-layer.

4 ~~4~~ Protocol Examples

4.1 ~~4.1~~ SOAP Requests

4.1.1 ~~4.1.1~~ SOAP Request with No Parameters

This example shows a request that does not specify the optional **Parameters** element.

```
POST /SqlBatch HTTP/1.1
Host: testServer
Content-Type: application/xml

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Body>
    <sql:sqlbatch>
      <sql:BatchCommands>
        SELECT @@version
      </sql:BatchCommands>
    </sql:sqlbatch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.1.2 ~~4.1.2~~ SOAP Request with SOAPAction Header

This example shows a request that specifies the SOAPAction header.

```
POST /SqlBatch HTTP/1.1
Host: testServer
Content-Type: application/xml
SOAPAction:
"http://schemas.microsoft.com/sqlserver/2004/SOAPsqlbatch"
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Body>
    <sql:sqlbatch>
      <sql:BatchCommands>
        SELECT @@version
      </sql:BatchCommands>
    </sql:sqlbatch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.1.3 ~~4.1.3~~ SOAP Request with Parameters

This example shows a request that specifies the **Parameters** element.

```
POST /SqlBatch HTTP/1.1
Content-Type: text/xml
Host: testServer
SOAPAction:
"http://schemas.microsoft.com/sqlserver/2004/SOAPsqlbatch"
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sqlparameter="http://schemas.microsoft.com/sqlserver/2004/
SOAP/types/SqlParameter">
```

```

xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Body>
    <sql:sqlbatch>
      <sql:BatchCommands>select @param1 as 'col1',
@param2 as
'col2'</sql:BatchCommands>
      <sql:Parameters>
        <sqlparameter:SqlParameter name="param1"
direction="InputOutput" sqlDbType="NVarChar" maxLength="10">
          <sqlparameter:Value>Hello World</sqlparameter:Value>
        </sqlparameter:SqlParameter>
        <sqlparameter:SqlParameter name="param2"
direction="Input" sqlDbType="VarChar">
          <sqlparameter:Value>hello</sqlparameter:Value>
        </sqlparameter:SqlParameter>
      </sql:Parameters>
    </sql:sqlbatch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.1.4 ~~4.1.4~~ SOAP Request with Additional Parameter Attributes

This example shows a request that specifies the parameter value instance type information.

```

POST /SqlBatch HTTP/1.1
Content-Type: text/xml
Host: testServer
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sqlparameter="http://schemas.microsoft.com/sqlserver/2004/
SOAP/types/SqlParameter"
xmlns:sqltypes=
"http://schemas.microsoft.com/sqlserver/2004/sqltypes"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Body>
    <sql:sqlbatch>
      <sql:BatchCommands>select @param1 as 'col1',
@param2 as
'col2'</sql:BatchCommands>
      <sql:Parameters>
        <sqlparameter:SqlParameter name="param1"
direction="InputOutput" sqlDbType="NVarChar" maxLength="10">
          <sqlparameter:Value xsi:type="sqltypes:varchar"
sqltypes:maxLength="15">Hello World</sqlparameter:Value>
        </sqlparameter:SqlParameter>
        <sqlparameter:SqlParameter name="param2"
direction="Input" maxLength="5" sqlDbType="VarChar">
          <sqlparameter:Value
xsi:type="sqltypes:int" sqltypes:int">100</sqlparameter:Value>
        </sqlparameter:SqlParameter>
      </sql:Parameters>
    </sql:sqlbatch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.1.5 ~~4.1.5~~ SOAP Request with sqlSession.initiate

This example shows a request to initiate a session.

```

POST /SqlBatch HTTP/1.1
Host: testServer

```

```

Content-Type:application/xml

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Header
xmlns:sqloptions=
"http://schemas.microsoft.com/sqlserver/
2004/SOAP/Options">
    <sqloptions:sqlSession initiate="true" timeout="12"/>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sql:sqlbatch>
      <sql:BatchCommands>
        Set language French
      </sql:BatchCommands>
    </sql:sqlbatch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.1.6 ~~4.1.6~~ SOAP Request with sqlSession.sessionId

This example shows a request specifying a query session to join.

```

POST /SqlBatch HTTP/1.1
Host: testServer
Content-Type:application/xml

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Header xmlns:sqloptions="http://schemas.microsoft.com/sqlserver/
2004/SOAP/Options">
    <sqloptions:sqlSession sessionId="P/Z+b7K3+UW/Xcc7ync2Yg==" />
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sql:sqlbatch>
      <sql:BatchCommands>
        SELECT @@language
      </sql:BatchCommands>
    </sql:sqlbatch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.1.7 ~~4.1.7~~ SOAP Request with sqlSession.terminate

This example shows a request to terminate a session.

```

POST /SqlBatch HTTP/1.1
Host: testServer
Content-Type:application/xml

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP">
  <SOAP-ENV:Header
xmlns:sqloptions="http://schemas.microsoft.com/sqlserver/
2004/SOAP/Options">
    <sqloptions:sqlSession terminate="true"
sessionId="P/Z+b7K3+UW/Xcc7ync2Yg==" />
  </SOAP-ENV:Header>

```

```

<SOAP-ENV:Body>
  <sql:sqlbatch>
    <sql:BatchCommands>
      SELECT @@language
    </sql:BatchCommands>
  </sql:sqlbatch>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2 ~~4.2~~—SOAP Responses

4.2.1 ~~4.2.1~~—SOAP Response with No Output Parameters

This example shows a response that does not specify any output for **SqlParameter** elements.

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/
sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <SOAP-ENV:Body>
    <sql:sqlbatchResponse>
      <sql:sqlbatchResult>
        <sqlresultstream:SqlRowSet
xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true"
msdata:UDTColumnValueWrapped="true">
          <xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/
2004/sqltypes">
            <xsd:simpleType name="nvarchar">
              <xsd:restriction base="xsd:string">
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:schema>
          </xsd:schema>
          targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet1"
          elementFormDefault="qualified">
            <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
              </xsd:import>
            <xsd:element name="SqlRowSet1"
msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">

```

```

        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Column1" minOccurs="0"
sql:sqlColumnName="">
              <xsd:simpleType>
                <xsd:restriction base="sqltypes:nvarchar"
sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                  <xsd:maxLength value="300">
                    </xsd:maxLength>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</diffgr:diffgram>
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  <SqlResultSet1
xmlns="urn:schemas-microsoft-com:sql:SqlResultSet1">
    <row>
      <Column1>Microsoft
SQL Server 2008 (RTM) - 10.0.1600.22 (X64)
Jul 9 2008 14:17:44
Copyright (c) 1988-2008 Microsoft Corporation
Developer Edition (64-bit) on
Windows NT 6.1 &lt;X64&gt; (Build 7000: )
</Column1>
    </row>
  </SqlResultSet1>
</diffgr:diffgram>
</sqlresultstream:SqlResultSet>
<sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
  <sqlrowcount:Count>1</sqlrowcount:Count>
</sqlresultstream:SqlRowCount>
</sql:sqlbatchResult>
</sql:sqlbatchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2.2 ~~4.2.2~~—SOAP Response with Output Parameters

This example shows a response that specifies output parameters in the **Parameters** element.

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlResultStream"

```

```

xmlns:sqltransaction="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/
2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <SOAP-ENV:Body>
    <sql:sqlbatchResponse>
      <sql:sqlbatchResult>
        <sqlresultstream:SqlRowSet xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true"
msdata:UDTColumnValueWrapped="true">
          <xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/
2004/sqltypes">
            <xsd:simpleType name="nvarchar">
              <xsd:restriction base="xsd:string">
                </xsd:restriction>
              </xsd:simpleType>
            <xsd:simpleType name="varchar">
              <xsd:restriction base="xsd:string">
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:schema>
          <xsd:schema
targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet1"
elementFormDefault="qualified">
            <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
              </xsd:import>
            <xsd:element name="SqlRowSet1" msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">
                    <xsd:complexType>
                      <xsd:sequence>
                        <xsd:element name="col1" minOccurs="0">
                          <xsd:simpleType>
                            <xsd:restriction base="sqltypes:nvarchar"
sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                              <xsd:maxLength value="10">
                                </xsd:maxLength>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        <xsd:element name="col2" minOccurs="0">
                          <xsd:simpleType>
                            <xsd:restriction base="sqltypes:varchar"
sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                              <xsd:maxLength value="1">
                                </xsd:maxLength>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:schema>
          </diffgr:diffgram
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">

```

```

        <SqlRowSet1
xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1">
    <row>
        <col1>Hello Worl</col1>
        <col2>h</col2>
    </row>
    </SqlRowSet1>
</diffgr:diffgram>
</sqlresultstream:SqlRowSet>
<sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
    <sqlrowcount:Count>1</sqlrowcount:Count>
</sqlresultstream:SqlRowCount>
</sql:sqlbatchResult>
<sql:Parameters xmlns:sqlparameter="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlParameter">
    <sqlparameter:SqlParameter name="param1" sqlDbType="NVarChar"
direction="InputOutput" maxLength="10">
        <sqlparameter:Value>Hello Worl</sqlparameter:Value>
    </sqlparameter:SqlParameter>
</sql:Parameters>
</sql:sqlbatchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2.3 ~~4.2.3~~ SOAP Response with Additional Output Parameter Attributes

This example shows a response that specifies the parameter value instance type information.

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/
2004/sqltypes" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <SOAP-ENV:Body>
        <sql:sqlbatchResponse>
            <sql:sqlbatchResult>
                <sqlresultstream:SqlRowSet xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true"
msdata:UDTColumnValueWrapped="true">
                    <xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/
2004/sqltypes">
                        <xsd:simpleType name="nvarchar">
                            <xsd:restriction base="xsd:string">
                                </xsd:restriction>
                            </xsd:simpleType>
                        <xsd:simpleType name="varchar">
                            <xsd:restriction base="xsd:string">
                                </xsd:restriction>
                            </xsd:simpleType>
                    </xsd:schema>
                </sqlresultstream:SqlRowSet>
            </sql:sqlbatchResult>
        </sql:sqlbatchResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```



```

    <xsd:schema
targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet1"
elementFormDefault="qualified">
  <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
  </xsd:import>
  <xsd:element name="SqlRowSet1" msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="col1" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="sqltypes:nvarchar"
sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                    <xsd:maxLength value="10">
                      </xsd:maxLength>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
              <xsd:element name="col2" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="sqltypes:varchar"
sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                    <xsd:maxLength value="5">
                      </xsd:maxLength>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  <diffgr:diffgram
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    <SqlRowSet1
xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1">
      <row>
        <col1>Hello Worl</col1>
        <col2>100</col2>
      </row>
    </SqlRowSet1>
  </diffgr:diffgram>
</sqlresultstream:SqlRowSet>
<sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
  <sqlrowcount:Count>1</sqlrowcount:Count>
</sqlresultstream:SqlRowCount>
</sql:sqlbatchResult>
  <sql:Parameters xmlns:sqlparameter="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlParameter">
    <sqlparameter:SqlParameter name="param1" sqlDbType="NVarChar"
direction="InputOutput" maxLength="10">
      <sqlparameter:Value xsi:type="sqltypes:varchar"
sqltypes:maxLength="15" sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">Hello Worl</sqlparameter:Value>
    </sqlparameter:SqlParameter>

```

```

    </sql:Parameters>
  </sql:sqlbatchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2.4 ~~4.2.4~~ SOAP Response to a Request with sqlSession.initiate

This example shows a response to an initiate session request.

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types" xmlns:sqlrowcount="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <SOAP-ENV:Header xmlns:sqloptions="http://schemas.microsoft.com/
sqlserver/2004/SOAP/Options">
    <sqloptions:sqlSession sessionId="P/Z+b7K3+UW/Xcc7ync2Yg=="
timeout="202">
      </sqloptions:sqlSession>
    </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sql:sqlbatchResponse>
      <sql:sqlbatchResult>
        </sql:sqlbatchResult>
      </sql:sqlbatchResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

```

4.2.5 ~~4.2.5~~ SOAP Response to a Request with sqlSession.sessionId

This example shows a response to a session join request.

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/

```

```

2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/
2004/sqltypes" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <SOAP-ENV:Header
xmlns:sqloptions="http://schemas.microsoft.com/sqlserver/
2004/SOAP/Options">
    <sqloptions:sqlSession sessionId="P/Z+b7K3+UW/Xcc7ync2Yg==">
    </sqloptions:sqlSession>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sql:sqlbatchResponse>
      <sql:sqlbatchResult>
        <sqlresultstream:SqlRowSet xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true"
msdata:UDTColumnValueWrapped="true">
          <xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/
2004/sqltypes">
            <xsd:simpleType name="nvarchar">
              <xsd:restriction base="xsd:string">
                </xsd:restriction>
            </xsd:simpleType>
          </xsd:schema>
          <xsd:schema
targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet1"
elementFormDefault="qualified">
            <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
              </xsd:import>
            <xsd:element name="SqlRowSet1" msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">
                    <xsd:complexType>
                      <xsd:sequence>
                        <xsd:element name="Column1" minOccurs="0"
sql:sqlColumnName="">
                          <xsd:simpleType>
                            <xsd:restriction
base="sqltypes:nvarchar" sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                              <xsd:maxLength value="128">
                                </xsd:maxLength>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:schema>
          </diffgr:diffgram
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
            <SqlRowSet1
xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1">
              <row>
                <Column1>Français</Column1>
              </row>
            </SqlRowSet1>
          </diffgr:diffgram>
        </sqlresultstream:SqlRowSet>
      <sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">

```

```

        <sqlrowcount:Count>1</sqlrowcount:Count>
    </sqlresultstream:SqlRowCount>
</sql:sqlbatchResult>
</sql:sqlbatchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2.6 ~~4.2.6~~ SOAP Response to a Request with sqlSession.terminate

This example shows a response to a terminate session request.

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/
sqlserver/2004/SOAP/types/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/
sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <SOAP-ENV:Header
xmlns:sqloptions="http://schemas.microsoft.com/sqlserver/
2004/SOAP/Options">
    <sqloptions:sqlSession terminate="true"
sessionId="P/Z+b7K3+UW/Xcc7ync2Yg==">
      </sqloptions:sqlSession>
    </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sql:sqlbatchResponse>
      <sql:sqlbatchResult>
        <sqlresultstream:SqlRowSet xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true"
msdata:UDTColumnValueWrapped="true">
          <xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/
2004/sqltypes">
            <xsd:simpleType name="nvarchar">
              <xsd:restriction base="xsd:string">
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:schema>
          <xsd:schema
targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet1"
elementFormDefault="qualified">
            <xsd:import namespace="http://schemas.microsoft.com/
sqlserver/2004/sqltypes">
              </xsd:import>
            <xsd:element name="SqlRowSet1" msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">

```

```

        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Column1" minOccurs="0"
sql:sqlColumnName="">
              <xsd:simpleType>
                <xsd:restriction
base="sqltypes:nvarchar"
sqltypes:localeId="1033"
sqltypes:sqlCompareOptions="IgnoreCase IgnoreWidth"
sqltypes:sqlCollationVersion="2">
                  <xsd:maxLength value="128">
                    </xsd:maxLength>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</diffgr:diffgram
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  <SqlResultSet1
xmlns="urn:schemas-microsoft-com:sql:SqlResultSet1">
    <row>
      <Column1>Français</Column1>
    </row>
  </SqlResultSet1>
</diffgr:diffgram>
</sqlresultstream:SqlResultSet>
<sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
  <sqlrowcount:Count>1</sqlrowcount:Count>
</sqlresultstream:SqlRowCount>
</sql:sqlbatchResult>
</sql:sqlbatchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2.7 ~~4.2.7~~ SOAP Fault Response

This example shows a sample SOAP fault response.

```

HTTP/1.1 500 Internal Server Error
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/
2004/SOAP/types/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/
sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">

```

```

    <SOAP-ENV:Body>
      <SOAP-ENV:Fault
        xmlns:sqlsoapfaultcode="http://schemas.microsoft.com/
sqlserver/2004/SOAP/SqlSoapFaultCode">
          <faultcode>SOAP-ENV:Client</faultcode>
          <faultstring>There was an error in the incoming
SOAP request packet: Client, SoapHeader,
SessionIdIsInvalid</faultstring>
          <faultactor>http://schemas.microsoft.com/sqlserver/
2004/SOAP</faultactor>
          <detail xmlns:SOAP-1_2-ENV="http://www.w3.org/
2003/05/soap-envelope">
            <SOAP-1_2-ENV:Code>
              <SOAP-1_2-ENV:Value>SOAP-1_2-ENV:Sender
</SOAP-1_2-ENV:Value>
              <SOAP-1_2-ENV:Subcode>
                <SOAP-1_2-
ENV:Value>sqlsoapfaultcode:SoapHeader</SOAP-1_2-ENV:Value>
                <SOAP-1_2-ENV:Subcode>
                  <SOAP-1_2-
ENV:Value>sqlsoapfaultcode:SessionIdIsInvalid</SOAP-1_2-ENV:Value>
                  </SOAP-1_2-ENV:Subcode>
                </SOAP-1_2-ENV:Subcode>
              </SOAP-1_2-ENV:Code>
              <SOAP-1_2-ENV:Reason>
                <SOAP-1_2-ENV:Text xml:lang="en-US">There was an error
in the incoming SOAP request packet: Sender, SoapHeader,
SessionIdIsInvalid</SOAP-1_2-ENV:Text>
              </SOAP-1_2-ENV:Reason>
              <SOAP-1_2-ENV:Node>http://server/endpointuri/
</SOAP-1_2-ENV:Node>
            </SOAP-1_2-
ENV:Role>http://schemas.microsoft.com/sqlserver/2004/SOAP</SOAP-
1_2-ENV:Role>
            <SOAP-1_2-ENV:Detail>
              </SOAP-1_2-ENV:Detail>
            </detail>
          </SOAP-ENV:Fault>
        </SOAP-ENV:Body>
      </SOAP-ENV:Envelope>

```

5 ~~5.1~~ Security

5.1 ~~5.1.1~~ Security Considerations for Implementers

This protocol assumes the underlying HTTP is secured using HTTPS. All security considerations are covered in sections 2 and 3.

5.2 ~~5.2.1~~ Index of Security Parameters

Security parameter	Section
WSSE-UsernameToken	3.1.4.3

6 ~~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:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:sqloptions="http://schemas.microsoft.com/sqlserver/2004/SOAP/Options"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP" xmlns:tns="http://tempuri.org"
targetNamespace="http://tempuri.org">
- <wsdl:types>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
attributeFormDefault="qualified"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">XML Schema describing the base types to which SQL Server
types are being mapped. For more information, please consult the documentation. (c) Copyright
2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML
format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have
trademarks, copyrights, or other intellectual property rights covering subject matter in the
schema. Microsoft does not make any representation or warranty regarding the schema or any
product or item developed based on the schema. The schema is provided to you on an AS IS
basis. Microsoft disclaims all express, implied and statutory warranties, including but not
limited to the implied warranties of merchantability, fitness for a particular purpose, and
freedom from infringement. Without limiting the generality of the foregoing, Microsoft does
not make any warranty of any kind that any item developed based on the schema, or any portion
of the schema, will not infringe any copyright, patent, trade secret, or other intellectual
property right of any person or entity in any country. It is your responsibility to seek
licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE
LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE
SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL
(INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN
A DIVISED OF SUCH DAMAGES.</xsd:documentation>
  </xsd:annotation>
- <!-- Global types and attributes that can be used for schema annotations.
-->
- <xsd:simpleType name="sqlDbTypeEnum">
- <xsd:restriction base="xsd:string">
  <xsd:enumeration value="BigInt" />
  <xsd:enumeration value="Binary" />
  <xsd:enumeration value="Bit" />
  <xsd:enumeration value="Char" />
  <xsd:enumeration value="DateTime" />
  <xsd:enumeration value="Decimal" />
  <xsd:enumeration value="Float" />
  <xsd:enumeration value="Image" />
  <xsd:enumeration value="Int" />
  <xsd:enumeration value="Money" />
  <xsd:enumeration value="NChar" />
  <xsd:enumeration value="NText" />
- <!-- The sqlDbTypeEnum aligns with the .Net System.Data.SqlDbType enum and does not
provide an entry for Numeric (which is mapped to Decimal).
-->
  <xsd:enumeration value="NVarChar" />
  <xsd:enumeration value="Real" />
  <xsd:enumeration value="SmallDateTime" />
  <xsd:enumeration value="SmallInt" />
  <xsd:enumeration value="SmallMoney" />
  <xsd:enumeration value="Text" />
  <xsd:enumeration value="Timestamp" />
  <xsd:enumeration value="TinyInt" />
  <xsd:enumeration value="Udt" />
  <xsd:enumeration value="UniqueIdentifier" />
  <xsd:enumeration value="VarBinary" />
  <xsd:enumeration value="VarChar" />
```



```

    <xsd:enumeration value="Variant" />
    <xsd:enumeration value="Xml" />
  </xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="sqlCompareOptionsEnum">
- <xsd:restriction base="xsd:string">
  <xsd:enumeration value="Default" />
  <xsd:enumeration value="None" />
  <xsd:enumeration value="IgnoreCase" />
  <xsd:enumeration value="IgnoreNonSpace" />
  <xsd:enumeration value="IgnoreKanaType" />
  <xsd:enumeration value="IgnoreWidth" />
  <xsd:enumeration value="BinarySort" />
  <xsd:enumeration value="BinarySort2" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="sqlCompareOptionsList">
  <xsd:list itemType="sqltypes:sqlCompareOptionsEnum" />
</xsd:simpleType>
  <xsd:attribute default="NVarChar" name="sqlDbType" type="sqltypes:sqlDbTypeEnum" />
  <xsd:attribute name="clrTypeName" type="xsd:string" />
  <xsd:attribute default="1" name="maxLength" type="xsd:long" />
  <xsd:attribute default="-1" name="localeId" type="xsd:int" />
  <xsd:attribute default="Default" name="sqlCompareOptions"
type="sqltypes:sqlCompareOptionsList" />
  <xsd:attribute default="0" name="sqlCollationVersion" type="xsd:int" />
  <xsd:attribute default="0" name="sqlSortId" type="xsd:int" />
  <xsd:attribute default="0" name="scale" type="xsd:unsignedByte" />
  <xsd:attribute default="18" name="precision" type="xsd:unsignedByte" />
  <xsd:attribute name="xmlSchemaCollection" type="xsd:string" />
  <xsd:attribute name="sqlTypeAlias" type="xsd:string" />
- <!-- Global types that describe the base SQL types.
-->
- <xsd:simpleType name="char">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
- <xsd:simpleType name="nchar">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
- <xsd:simpleType name="varchar">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
- <xsd:simpleType name="nvarchar">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
- <xsd:simpleType name="text">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
- <xsd:simpleType name="ntext">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
- <xsd:simpleType name="varbinary">
  <xsd:restriction base="xsd:base64Binary" />
</xsd:simpleType>
- <xsd:simpleType name="binary">
  <xsd:restriction base="xsd:base64Binary" />
</xsd:simpleType>
- <xsd:simpleType name="image">
  <xsd:restriction base="xsd:base64Binary" />
</xsd:simpleType>
- <xsd:simpleType name="timestamp">
- <xsd:restriction base="xsd:base64Binary">
  <xsd:maxLength value="8" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="timestampNumeric">
- <!-- The timestampNumeric type supports a legacy format of timestamp.
-->
  <xsd:restriction base="xsd:long" />

```

```

    </xsd:simpleType>
- <xsd:simpleType name="decimal">
  <xsd:restriction base="xsd:decimal" />
</xsd:simpleType>
- <xsd:simpleType name="numeric">
  <xsd:restriction base="xsd:decimal" />
</xsd:simpleType>
- <xsd:simpleType name="bigint">
  <xsd:restriction base="xsd:long" />
</xsd:simpleType>
- <xsd:simpleType name="int">
  <xsd:restriction base="xsd:int" />
</xsd:simpleType>
- <xsd:simpleType name="smallint">
  <xsd:restriction base="xsd:short" />
</xsd:simpleType>
- <xsd:simpleType name="tinyint">
  <xsd:restriction base="xsd:unsignedByte" />
</xsd:simpleType>
- <xsd:simpleType name="bit">
  <xsd:restriction base="xsd:boolean" />
</xsd:simpleType>
- <xsd:simpleType name="float">
  <xsd:restriction base="xsd:double" />
</xsd:simpleType>
- <xsd:simpleType name="real">
  <xsd:restriction base="xsd:float" />
</xsd:simpleType>
- <xsd:simpleType name="datetime">
- <xsd:restriction base="xsd:dateTime">
  <xsd:pattern value="((000[1-9])|(00[1-9][0-9])|(0[1-9][0- 9]{2})|([1-9][0-9]{3}))-(0[1-9])|(1[012]))-((0[1-9])|([12][0- 9])|(3[01]))T((([01][0-9])|(2[0-3]))(:[0-5][0-9]){2}(\.[0-9]{2}[037])?)?"/>
  <xsd:maxInclusive value="9999-12-31T23:59:59.997" />
  <xsd:minInclusive value="1753-01-01T00:00:00.000" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="smalldatetime">
- <xsd:restriction base="xsd:dateTime">
  <xsd:pattern value="((000[1-9])|(00[1-9][0-9])|(0[1-9][0- 9]{2})|([1-9][0-9]{3}))-(0[1-9])|(1[012]))-((0[1-9])|([12][0- 9])|(3[01]))T((([01][0-9])|(2[0-3]))(:[0-5][0-9])?(:00)"/>
  <xsd:maxInclusive value="2079-06-06T23:59:00" />
  <xsd:minInclusive value="1900-01-01T00:00:00" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="money">
- <xsd:restriction base="xsd:decimal">
  <xsd:totalDigits value="19" />
  <xsd:fractionDigits value="4" />
  <xsd:maxInclusive value="922337203685477.5807" />
  <xsd:minInclusive value="-922337203685477.5808" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="smallmoney">
- <xsd:restriction base="xsd:decimal">
  <xsd:totalDigits value="10" />
  <xsd:fractionDigits value="4" />
  <xsd:maxInclusive value="214748.3647" />
  <xsd:minInclusive value="-214748.3648" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:simpleType name="uniqueidentifier">
- <xsd:restriction base="xsd:string">
  <xsd: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})|(\{([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})\})"/>
</xsd:restriction>
</xsd:simpleType>
- <!-- sql_variant directly maps to xsd:anyType

```

```

-->
- <xsd:complexType name="xml" mixed="true">
- <xsd:sequence>
  <xsd:any minOccurs="0" maxOccurs="unbounded" processContents="skip" />
</xsd:sequence>
</xsd:complexType>
- <!-- the following type is for FOR XML binary URL results only
-->
- <xsd:simpleType name="dbobject">
  <xsd:restriction base="xsd:anyURI" />
</xsd:simpleType>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
schema for Microsoft SQL Server is presented in XML format and is for informational purposes
only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
intellectual property rights covering subject matter in the schema. Microsoft does not make
any representation or warranty regarding the schema or any product or item developed based on
the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,
implied and statutory warranties, including but not limited to the implied warranties of
merchantability, fitness for a particular purpose, and freedom from infringement. Without
limiting the generality of the foregoing, Microsoft does not make any warranty of any kind
that any item developed based on the schema, or any portion of the schema, will not infringe
any copyright, patent, trade secret, or other intellectual property right of any person or
entity in any country. It is your responsibility to seek licenses for such intellectual
property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND
ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY
DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL
DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType name="SqlRowSet">
- <xsd:sequence maxOccurs="unbounded">
  <xsd:element ref="xsd:schema" />
  <xsd:any />
</xsd:sequence>
</xsd:complexType>
- <xsd:complexType name="SqlXml" mixed="true">
- <xsd:sequence>
  <xsd:any processContents="skip" />
</xsd:sequence>
</xsd:complexType>
- <xsd:simpleType name="SqlResultCode">
- <xsd:restriction base="xsd:int">
  <xsd:minInclusive value="0" />
</xsd:restriction>
</xsd:simpleType>
  <xsd:attribute name="IsDataSetWithSchema" type="xsd:boolean" />
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlTransaction">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
schema for Microsoft SQL Server is presented in XML format and is for informational purposes
only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
intellectual property rights covering subject matter in the schema. Microsoft does not make
any representation or warranty regarding the schema or any product or item developed based on
the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,
implied and statutory warranties, including but not limited to the implied warranties of
merchantability, fitness for a particular purpose, and freedom from infringement. Without
limiting the generality of the foregoing, Microsoft does not make any warranty of any kind
that any item developed based on the schema, or any portion of the schema, will not infringe
any copyright, patent, trade secret, or other intellectual property right of any person or
entity in any country. It is your responsibility to seek licenses for such intellectual
property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND
ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY

```

```

DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL
DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>
  </xsd:annotation>
- <xsd:complexType name="SqlTransaction">
- <xsd:sequence minOccurs="1" maxOccurs="1">
  <xsd:element name="Descriptor" type="xsd:base64Binary" />
- <xsd:element name="Type">
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
  <xsd:enumeration value="Begin" />
  <xsd:enumeration value="Commit" />
  <xsd:enumeration value="Rollback" />
  <xsd:enumeration value="EnlistDTC" />
  <xsd:enumeration value="Defect" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
elementFormDefault="qualified" targetNamespace="http://schemas.microsoft.com/sqlserver/2004/
SOAP/types/SqlRowCount">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
schema for Microsoft SQL Server is presented in XML format and is for informational purposes
only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
intellectual property rights covering subject matter in the schema. Microsoft does not make
any representation or warranty regarding the schema or any product or item developed based on
the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,
implied and statutory warranties, including but not limited to the implied warranties of
merchantability, fitness for a particular purpose, and freedom from infringement. Without
limiting the generality of the foregoing, Microsoft does not make any warranty of any kind
that any item developed based on the schema, or any portion of the schema, will not infringe
any copyright, patent, trade secret, or other intellectual property right of any person or
entity in any country. It is your responsibility to seek licenses for such intellectual
property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND
ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY
DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL
DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>
  </xsd:annotation>
- <xsd:complexType name="SqlRowCount">
- <xsd:sequence minOccurs="1" maxOccurs="1">
  <xsd:element name="Count" type="xsd:long" />
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/2004/ SOAP/types/SqlMessage"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlMessage">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
schema for Microsoft SQL Server is presented in XML format and is for informational purposes
only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
intellectual property rights covering subject matter in the schema. Microsoft does not make
any representation or warranty regarding the schema or any product or item developed based on
the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,
implied and statutory warranties, including but not limited to the implied warranties of
merchantability, fitness for a particular purpose, and freedom from infringement. Without
limiting the generality of the foregoing, Microsoft does not make any warranty of any kind
that any item developed based on the schema, or any portion of the schema, will not infringe
any copyright, patent, trade secret, or other intellectual property right of any person or
entity in any country. It is your responsibility to seek licenses for such intellectual
property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND
ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY
DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL
DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>
  </xsd:annotation>
- <xsd:simpleType name="nonNegativeInteger">

```

```

- <xsd:restriction base="xsd:int">
  <xsd:minInclusive value="0" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:complexType name="SqlMessage">
- <xsd:sequence minOccurs="1" maxOccurs="1">
  <xsd:element name="Class" type="sqlmessage:nonNegativeInteger" />
  <xsd:element name="LineNumber" type="sqlmessage:nonNegativeInteger" />
  <xsd:element name="Message" type="xsd:string" />
  <xsd:element name="Number" type="sqlmessage:nonNegativeInteger" />
  <xsd:element name="Procedure" type="xsd:string" minOccurs="0" />
  <xsd:element name="Server" type="xsd:string" minOccurs="0" />
  <xsd:element name="Source" type="xsd:string" />
  <xsd:element name="State" type="sqlmessage:nonNegativeInteger" />
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/2004/SOAP/types"
  xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlMessage"
  xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlRowCount"
  xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlTransaction"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlResultStream">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
  schema for Microsoft SQL Server is presented in XML format and is for informational purposes
  only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
  intellectual property rights covering subject matter in the schema. Microsoft does not make
  any representation or warranty regarding the schema or any product or item developed based on
  the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,
  implied and statutory warranties, including but not limited to the implied warranties of
  merchantability, fitness for a particular purpose, and freedom from infringement. Without
  limiting the generality of the foregoing, Microsoft does not make any warranty of any kind
  that any item developed based on the schema, or any portion of the schema, will not infringe
  any copyright, patent, trade secret, or other intellectual property right of any person or
  entity in any country. It is your responsibility to seek licenses for such intellectual
  property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND
  ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY
  DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL
  DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>
  </xsd:annotation>
  <xsd:import namespace="http://schemas.microsoft.com/ sqlserver/2004/SOAP/types" />
  <xsd:import namespace="http://schemas.microsoft.com/ sqlserver/2004/SOAP/types/SqlMessage"
  />
  <xsd:import namespace="http://schemas.microsoft.com/ sqlserver/2004/SOAP/types/SqlRowCount"
  />
  <xsd:import namespace="http://schemas.microsoft.com/
  sqlserver/2004/SOAP/types/SqlTransaction" />
- <xsd:complexType name="SqlResultStream">
- <xsd:choice minOccurs="1" maxOccurs="unbounded">
  <xsd:element name="SqlRowSet" type="sqlsoaptypes:SqlRowSet" />
  <xsd:element name="SqlXml" type="sqlsoaptypes:SqlXml" />
  <xsd:element name="SqlMessage" type="sqlmessage:SqlMessage" />
  <xsd:element name="SqlRowCount" type="sqlrowcount:SqlRowCount" />
  <xsd:element name="SqlResultCode" type="sqlsoaptypes:SqlResultCode" />
  <xsd:element name="SqlTransaction" type="sqltransaction:SqlTransaction" />
</xsd:choice>
</xsd:complexType>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
  elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/Options">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
  schema for Microsoft SQL Server is presented in XML format and is for informational purposes
  only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
  intellectual property rights covering subject matter in the schema. Microsoft does not make
  any representation or warranty regarding the schema or any product or item developed based on

```

the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>

```
</xsd:annotation>
- <xsd:element name="initialDatabase">
- <xsd:annotation>
  <xsd:documentation>Set initial database on login.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The name of the initial database to attach to.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="optional" default="false" type="xsd:boolean" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Whether the initial database is optional or not.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="filename" type="xsd:string" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>The filename of the database to attach to.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="initialLanguage">
- <xsd:annotation>
  <xsd:documentation>Set initial language to set.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The name of the initial language to set.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="optional" default="false" type="xsd:boolean" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Whether the initial language is optional or not.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="environmentChangeNotifications">
- <xsd:annotation>
  <xsd:documentation>Receive environment change notifications.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="databaseChange" default="false" type="xsd:boolean" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Receive notifications of database changes.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="languageChange" default="false" type="xsd:boolean" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Receive notifications of language changes.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="transactionBoundary" default="false" type="xsd:boolean"
form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Receive notifications of transaction boundaries.</xsd:documentation>
```

```

    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="applicationName">
- <xsd:annotation>
  <xsd:documentation>Set the application name for the login.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The application name to set for the login.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="hostName">
- <xsd:annotation>
  <xsd:documentation>Set the host name for the login.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The host name to set for the login.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="clientPID">
- <xsd:annotation>
  <xsd:documentation>Set the client process ID for the login.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:long" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The client process ID to set for the login.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="clientNetworkID">
- <xsd:annotation>
  <xsd:documentation>Set the client network ID for the login.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:base64Binary" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The client network ID to set for the login.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="clientInterface">
- <xsd:annotation>
  <xsd:documentation>Set the client interface for the login.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="value" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The client interface to set for the login.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="notificationRequest">
- <xsd:annotation>
  <xsd:documentation>Requests query notifications for the request.</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>

```

```

- <xsd:attribute name="notificationId" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The notification identifier.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="deliveryService" type="xsd:string" form="unqualified" use="required">
- <xsd:annotation>
  <xsd:documentation>The delivery service.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="timeout" type="xsd:integer" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>The timeout value.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
- <xsd:element name="sqlSession">
- <xsd:annotation>
  <xsd:documentation>SQL Server SOAP Session</xsd:documentation>
</xsd:annotation>
- <xsd:complexType>
- <xsd:attribute name="initiate" default="false" type="xsd:boolean" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Set to 'true' to request to start a new session.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="terminate" default="false" type="xsd:boolean" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>Set to 'true' to request to terminate an existing
session.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="sessionId" type="xsd:base64Binary" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>The ID of a session.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="timeout" type="xsd:int" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>The timeout in seconds before the session expires.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="transactionDescriptor" type="xsd:base64Binary" form="unqualified">
- <xsd:annotation>
  <xsd:documentation>The descriptor of a transaction to enlist to.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sqlparameter="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"
elementFormDefault="qualified" attributeFormDefault="qualified">
- <xsd:annotation>
  <xsd:documentation xml:lang="en">(c) Copyright 2004, Microsoft Corporation The following
schema for Microsoft SQL Server is presented in XML format and is for informational purposes
only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other
intellectual property rights covering subject matter in the schema. Microsoft does not make
any representation or warranty regarding the schema or any product or item developed based on
the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,
implied and statutory warranties, including but not limited to the implied warranties of
merchantability, fitness for a particular purpose, and freedom from infringement. Without
limiting the generality of the foregoing, Microsoft does not make any warranty of any kind
that any item developed based on the schema, or any portion of the schema, will not infringe
any copyright, patent, trade secret, or other intellectual property right of any person or
entity in any country. It is your responsibility to seek licenses for such intellectual

```


property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.</xsd:documentation>

```
</xsd:annotation>
  <xsd:import namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes" />
- <xsd:simpleType name="ParameterDirection">
- <xsd:restriction base="xsd:string">
  <xsd:enumeration value="Input" />
  <xsd:enumeration value="InputOutput" />
</xsd:restriction>
</xsd:simpleType>
- <xsd:complexType name="ArrayOfSqlParameter">
- <xsd:sequence>
  <xsd:element minOccurs="0" maxOccurs="unbounded" name="SqlParameter"
type="sqlparameter:SqlParameter" />
</xsd:sequence>
</xsd:complexType>
- <xsd:complexType name="SqlParameter">
- <xsd:sequence>
  <xsd:element minOccurs="1" maxOccurs="1" name="Value" nillable="true" />
</xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required" form="unqualified" />
  <xsd:attribute default="NVarChar" name="sqlDbType" type="sqltypes:sqlDbTypeEnum"
use="optional" form="unqualified" />
  <xsd:attribute default="Input" name="direction" type="sqlparameter:ParameterDirection"
use="optional" form="unqualified" />
  <xsd:attribute default="1" name="maxLength" type="xsd:long" use="optional"
form="unqualified" />
  <xsd:attribute default="18" name="precision" type="xsd:unsignedByte" use="optional"
form="unqualified" />
  <xsd:attribute default="0" name="scale" type="xsd:unsignedByte" use="optional"
form="unqualified" />
  <xsd:attribute default="" name="clrTypeName" type="xsd:string" use="optional"
form="unqualified" />
  <xsd:attribute default="Default" name="sqlCompareOptions"
type="sqltypes:sqlCompareOptionsList" use="optional" form="unqualified" />
  <xsd:attribute default="-1" name="localeId" type="xsd:int" use="optional"
form="unqualified" />
  <xsd:attribute default="0" name="sqlCollationVersion" type="xsd:int" use="optional"
form="unqualified" />
  <xsd:attribute default="0" name="sqlSortId" type="xsd:int" use="optional"
form="unqualified" />
  <xsd:attribute default="" name="xmlSchemaCollection" type="xsd:string" use="optional"
form="unqualified" />
</xsd:complexType>
</xsd:schema>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlResultStream"
xmlns:sqlparameter="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter">
  <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlResultStream" />
  <xsd:import namespace="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlParameter"
/>
- <xsd:element name="sqlbatch">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element minOccurs="1" maxOccurs="1" name="BatchCommands" type="xsd:string" />
  <xsd:element minOccurs="0" maxOccurs="1" name="Parameters"
type="sqlparameter:ArrayOfSqlParameter" nillable="true" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
- <xsd:element name="sqlbatchResponse">
```

```

- <xsd:complexType>
- <xsd:sequence>
  <xsd:element minOccurs="1" maxOccurs="1" name="sqlbatchResult"
type="sqlresultstream:SqlResultStream" nillable="false" />
  <xsd:element minOccurs="0" maxOccurs="1" name="Parameters"
type="sqlparameter:ArrayOfSqlParameter" nillable="true" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
- <wsdl:message name="sqlbatchSoapIn">
  <wsdl:part name="parameters" element="sql:sqlbatch" />
</wsdl:message>
- <wsdl:message name="sqlbatchSoapOut">
  <wsdl:part name="parameters" element="sql:sqlbatchResponse" />
</wsdl:message>
- <wsdl:portType name="Batch_EPSoap">
- <wsdl:operation name="sqlbatch">
  <wsdl:input message="tns:sqlbatchSoapIn" />
  <wsdl:output message="tns:sqlbatchSoapOut" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="Batch_EPSoap" type="tns:Batch_EPSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="sqlbatch">
  <soap:operation soapAction="http://schemas.microsoft.com/sqlserver/2004/SOAPsqlbatch"
style="document" />
- <wsdl:input>
  <soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="Batch_EP">
- <wsdl:port name="Batch_EP" binding="tns:Batch_EPSoap">
  <soap:address location="http://testServer/SqlBatch" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

7 ~~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~~ Microsoft SQL Server 2005
- ~~Microsoft~~ Microsoft SQL Server 2008
- ~~Microsoft~~ Microsoft SQL Server 2008 R2

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 2.2.2.2.1.1.3: The possible values of **SqlMessage.Source** are listed in the following table.

Value	Meaning
Microsoft-SQL/9.0	This value is returned as part of SqlMessage.Source when the SqlMessage is generated by SQL Server 2005.
Microsoft-SQL/10.0	This value is returned as part of SqlMessage.Source when the SqlMessage is generated by Microsoft SQL Server 2008.

<2> Section 2.2.4.1.1: The supported values for **SqlParameter.sqlCollationVersion** are listed in the following table.

Value	Meaning
0	Microsoft SQL Server 2000 collation
1	SQL Server 2005 collation
2	SQL Server 2008 collation

<3> Section 2.2.5.1: The meanings for BinarySort and BinarySort2 in **sqlCompareOptionsEnum** are listed in the following table.

Value	Meaning
BinarySort	Sorts and compares string data based on the bit patterns defined for each character that is compatible with SQL Server 2000 binary sorting.
BinarySort2	Sorts and compares string data based on code-point comparison semantics in SQL Server 2005.

<4> Section 3.1.3: For more information about establishing additional listening endpoints, see [MSDN-SSLNXWS].

8 ~~8~~ Change Tracking

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

9 ~~9~~—Index

A

[Abstract data model](#) [42](#)
[_server](#) [42](#)
[Applicability](#) [14](#)
Applicability statement [14](#)
[applicationName](#) [20](#)
ArrayOfSqlParameter [31](#)
 complex type [31](#)
[ArrayOfSqlParameter complex type](#) [31](#)
[Attribute groups](#) [41](#)
[Attributes](#) [41](#)
authentication [49](#)

~~48~~

B

Batch_EPSoap server details [42](#)

~~42~~

C

Capability negotiation [14](#)
Change tracking [76](#)
[clientInterface](#) [20](#)
[clientNetworkID](#) [20](#)
[clientPID](#) [20](#)
Common message syntax [15](#)
Complex type
 ArrayOfSqlParameter [31](#)
Complex types [31](#)
 [ArrayOfSqlParameter](#) [31](#)

D

[Data model - abstract](#)
[_server](#) [42](#)
Directory service schema elements [41](#)

~~41~~

E

Elements [30](#)
[Elements - directory service schema](#) [41](#)
environmentChangeNotifications [21](#)
[Events](#)
 [_local - server](#) [49](#)
 [_timer - server](#) [49](#)

Examples
 protocol [50](#)

~~50~~

F

Fields
 vendor-extensible [14](#)
[Fields - vendor-extensible](#) [14](#)
Full WSDL [64](#)

~~66~~

G

Glossary [7](#)
[Groups](#) [41](#)

H

hostName 22

~~20~~

I

[Implementer - security considerations](#) 63

[Index of security parameters](#) 63

Informative references 11~~9~~

initialDatabase 22~~20~~

[Initialization](#)

[_server](#) 42

initialLanguage 22~~21~~

Introduction 7~~7~~

L

[Local events](#)

[_server](#) 49

M

[Message processing](#)

[_server](#) 43

Message syntax

common 15~~13~~

Messages 15~~13~~

[_ArrayOfSqlParameter complex type](#) 31

[_attribute groups](#) 41

[_attributes](#) 41

[_complex types](#) 31

[_elements](#) 30

[_enumerated](#) 16

[_groups](#) 41

[_namespaces](#) 15

[_simple types](#) 34

[_sqlbatchSoapIn](#) 16

sqlbatchSoapIn and sqlbatchSoapOut 16~~14~~

[_sqlbatchSoapIn message](#) 16

[_sqlbatchSoapOut](#) 24

[_sqlbatchSoapOut message](#) 24

[_sqlCompareOptionsList simple type](#) 35

[_sqlDbTypeEnum simple type](#) 39

[_sqlTypes simple type](#) 36

[_syntax](#) 15

transport 15

~~13~~

N

Namespaces 15~~13~~

Normative references 10~~8~~

notificationRequest 23

~~21~~

O

[Operations](#)

[_Authentication](#) 49

[_Session-based sqlbatch](#) 46

[_Single sqlbatch](#) 43

Overview 11~~9~~

[Overview \(synopsis\)](#) 11

P

[Parameters - security index](#) 63

[Preconditions](#) 13

[Prerequisites](#) 13
Prerequisites/preconditions 13~~11~~
Product behavior 75~~79~~
Protocol ~~details~~ [Details](#) 42
[overview](#) 42
Protocol examples 50~~50~~
 SOAP requests 50~~50~~
 SOAP responses 53

~~53~~

R

References 9
 informative 11~~9~~
 normative 10~~8~~
 [normative and informative](#) 8
Relationship to other protocols 12

~~10~~

S

[Schema elements - directory service](#) 41
Security 63~~65~~
 [implementer considerations](#) 63
 [parameter index](#) 63
[Sequencing rules](#)
 [server](#) 43
[Server](#)
 [abstract data model](#) 42
 [Authentication operation](#) 49
 [initialization](#) 42
 [local events](#) 49
 [message processing](#) 43
 [sequencing rules](#) 43
 Session-based sqlbatch ~~45~~ [operation](#) 46
 [Single sqlbatch operation](#) 43
 [timer events](#) 49
 [timers](#) 42
[Session-based sqlbatch](#) 46
Simple type
 sqlCompareOptionsList 35~~34~~
 sqlTypes 36~~35~~
Simple types 34~~34~~
 [sqlCompareOptionsList](#) 35
 [sqlDbTypeEnum](#) 39
 [sqlTypes](#) 36
Single sqlbatch 43~~43~~
SOAP requests
 examples 50~~50~~
SOAP responses
 examples 53~~53~~
sqlbatch
 session-based 46~~45~~
 single 43~~43~~
sqlbatchSoapIn (section 2.2.2 16~~14~~, section 2.2.2.1 16~~14~~)
sqlbatchSoapIn SOAP body 16~~14~~
sqlbatchSoapIn SOAP headers 17~~15~~
sqlbatchSoapOut (section 2.2.2 16~~14~~, section 2.2.2.2 24~~22~~)
sqlbatchSoapOut SOAP body 24~~23~~
sqlbatchSoapOut SOAP header 29~~28~~
sqlCompareOptionsList simple type 35~~34~~
sqlDbTypeEnum 39~~39~~
[sqlDbTypeEnum simple type](#) 39
SqlMessage 27~~26~~
SqlOptions 15~~13~~
SqlParameter 32~~31~~
SqlParameter.Value 34~~33~~

SqlResultCode 28~~27~~
SqlResultStream 44~~44~~
SqlRowCount 28~~27~~
SqlRowSet 25~~24~~
sqlSession (section 2.2.2.1.2.10 23~~22~~, section 3.1.4.2.2.1 47~~47~~)
sqlSession SOAP header 30~~29~~
SqlTransaction 28~~27~~
sqlTypes simple type 36~~35~~
SqlXml 26~~25~~
Standards assignments 14~~12~~
[Syntax](#)
[_messages - overview](#) 15

T

[Timer events](#)
[_server](#) 49
[Timers](#)
[_server](#) 42
Tracking changes 76~~81~~
Transport 15
[Types](#)
[_complex](#) 31~~13~~
[_simple](#) 34

V

Vendor-extensible fields 14~~12~~
Versioning 14
~~12~~

W

WSDL 64~~66~~
WSSUTP (section 2.2.2.1 16~~14~~, section 3.1.4.3 49~~48~~)