Advanced Message Queuing Protocol (AMQP) WebSocket Binding (WSB) Version 1.0

Committee Specification Draft 01

22 April 2014

Specification URIs

This version:

http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/csd01/amqp-wsb-v1.0-csd01.doc (Authoritative)
http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/csd01/amqp-wsb-v1.0-csd01.html
http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/csd01/amqp-wsb-v1.0-csd01.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/amqp-wsb-v1.0.doc (Authoritative)
http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/amqp-wsb-v1.0.html
http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/amqp-wsb-v1.0.pdf

Technical Committee:

OASIS Advanced Message Queuing Protocol (AMQP) Bindings and Mappings (AMQP-BINDMAP) TC

Chair:

Steve Huston (shuston@riverace.com), Individual

Editors:

David Ingham (dingham@redhat.com), Red Hat
Robert Godfrey (robert.godfrey@jpmorgan.com), JPMorgan Chase Bank, N.A.

Related work:

This specification is related to:

- *OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0 Part 0: Overview*. Edited by Robert Godfrey, David Ingham, and Rafael Schloming. 29 October 2012. OASIS Standard. http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html.

Abstract:

The AMQP WebSocket Binding specification defines a mechanism for tunneling an AMQP connection over a WebSocket transport. It is applicable as an approach for general firewall tunneling and for Web browser messaging scenarios.

Status:

This document was last revised or approved by the OASIS Advanced Message Queuing Protocol (AMQP) Bindings and Mappings (AMQP-BINDMAP) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at https://www.oasis-open.org/committees/amqp-bindmap/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (https://www.oasis-open.org/committees/amqp-bindmap/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[amqp-wsb-v1.0]

Advanced Message Queuing Protocol (AMQP) WebSocket Binding (WSB) Version 1.0. Edited by David Ingham and Robert Godfrey. 22 April 2014. OASIS Committee Specification Draft 01. http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/csd01/amqp-wsb-v1.0-csd01.html. Latest version: http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/amqp-wsb-v1.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

This specification describes how the WebSocket protocol can be used as a transport for AMQP 1.0 protocol traffic. It is applicable for two main scenarios:

- **Firewall traversal**. Since Websocket connection establishment is implemented as standard HTTP traffic using the default ports (80 and 443), it is often able to pass through network security devices without requiring special configuration or opening of additional ports. In this scenario, the AMQP communication can be between arbitrary AMQP peers, e.g., between an application using an AMQP client library and a message broker.
- **Browser-based messaging**. Since many Web browsers have built-in support for the WebSocket protocol, then this binding can be used to enable AMQP messaging through to the browser thereby enable a broad range of browser-based messaging scenarios.

An AMQP 1.0 protocol session begins with the protocol version negotiation, including the establishment of layered security layers if required. Once this is complete, the session continues with the exchange of AMQP frames that control the operation of the protocol and transport the business messages between the communicating peers.

The remainder of this specification describes how these concepts are mapped to a WebSocket transport.

1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

1.2 Normative References

[AMQP]	Godfrey, Robert; Ingham, David; Schloming, Rafael, “Advanced Message Queuing Protocol (AMQP) Version 1.0”, October 2012. OASIS Standard. http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.
[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., "Hypertext Transfer Protocol -- HTTP/1.1", RFC2616, June 1999. http://tools.ietf.org/html/rfc2616.
[RFC4422]	Melnikov, A., and Zeilenga, K., “Simple Authentication and Security Layer (SASL)", RFC4422, June 2006. http://tools.ietf.org/html/rfc4422.
[RFC6455]	Fette, I., and Melinkov, A., “The WebSocket Protocol”, December 2011. RFC 6455, December 2011. http://tools.ietf.org/html/rfc6455**.**

2 Opening a Connection

Opening a protocol session using the AMQP WebSocket binding involves the following sequence of steps:

- Opening a WebSocket connection
- Performing the AMQP protocol version handshake including optional establishment of a SASL security layer **[RFC4422]**
- Exchanging AMQP frames

These steps are described in the following subsections.

2.1 Opening a WebSocket Connection

The WebSocket Protocol connection MUST be opened as described in Section 4 of the WebSocket specification **[RFC6455]**. The initiating AMQP endpoint (the WebSocket Client) sends a HTTP GET request to the receiving AMQP endpoint (the WebSocket Server) identifying AMQP 1.0 as the subprotocol being used.

The table below defines how the elements of this HTTP WebSocket upgrade request are used in the context of the AMQP WebSocket binding:

WebSocket request element	**Usage in the AMQP WebSocket binding**
Request-URI of the HTTP GET request	This binding does not define any semantic meaning to this field. An implementation MAY interpret this field in an implementation-specific manner.
Host HTTP header	Identifies the hostname of the AMQP container. The value provided here SHOULD match that provided later in the hostname field of the `open` frame, and during SASL negotiation (if used).
Sec-WebSocket-Protocol HTTP header	Identifies the WebSocket subprotocol. For this AMQP WebSocket binding, the value MUST be set to the US-ASCII text string "`AMQPWSB10`" which refers to the 1.0 version of the AMQP Web Socket Binding as defined in this document.

As per the WebSocket specification **[RFC6455]**, if the Server agrees to the WebSocket upgrade to the requested subprotocol then it MUST respond with an HTTP Status-Line with status code 101 ("Switching Protocols") and echo the requested subprotocol in the Sec-WebSocket-Protocol HTTP header. Alternatively, the Server MAY return a redirect response (HTTP 3XX) and/or request HTTP level authentication. Section 4.2.2 of the Websocket specification **[RFC6455]** provides full details on the permutations of the Server's opening handshake.

If the Client does not receive a response with HTTP status code 101 and an HTTP Sec-WebSocket-Protocol equal to the US-ASCII text string "AMQPWSB10" then the Client MUST close the socket connection. If the Client receives a HTTP 3XX redirect response from the Server, the Client MAY follow the redirect and attempt connection establishment at the provided endpoint. If the response indicates a request for HTTP-level authentication then the Client MAY attempt connection establishment with new authentication credentials. Note that this specification does not define any relationship between any such HTTP-level credentials and the credentials that MAY be later provided as part of a SASL protocol layer negotiation. However, a particular implementation MAY impose some relationship.

An example of a successful upgrade handshake is shown below in Section 2.1.1.

2.1.1 Example

This section is non-normative.

An example WebSocket upgrade handshake to the AMQP WebSocket binding subprotocol is shown below.

The request from the client looks as follows:

```
GET /examplepath HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Protocol: AMQPWSB10
Sec-WebSocket-Version: 13
```

The response from the server looks as follows:

```
   HTTP/1.1 101 Switching Protocols
   Upgrade: websocket
   Connection: Upgrade
   Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
   Sec-WebSocket-Protocol: AMQPWSB10
```

2.2 Establishing a SASL Security Layer

AMQP peers MAY choose to establish a SASL security layer prior to commencing the raw AMQP protocol as defined in Part 5 of the AMQP 1.0 Specification **[AMQP]**. If SASL is not required then this section is omitted and the connection establishment process continues as defined in the Section below entitled AMQP Protocol Version Handshake.

Note that SASL mechanisms that provide for encryption in addition to authentication are not supported with this AMQP WebSocket binding.

2.2.1 SASL Protocol Header

If a SASL security layer is required then each peer MUST start by sending a protocol header. The protocol header consists of the upper case ASCII letters “AMQP” followed by a protocol id of three, followed by three unsigned bytes representing the major, minor, and revision of the specification version (currently 1 (SASL-MAJOR), 0 (SASLMINOR),0 (SASL-REVISION)). In total this is an 8-octet sequence:

4 octets	1 octet	1 octet	1 octet	1 octet
“AMQP”	%d3	major	minor	revision

This protocol header MUST be sent as the contents of a single WebSocket message.

2.2.2 SASL Negotiation

The SASL negotiation is implemented by the exchange of SASL frames as defined in Section 5.3 of the AMQP 1.0 specification **[AMQP]**. Each SASL frame MUST be transferred as a WebSocket message.

After successful establishment of the SASL layer then the connection establishment process continues with the AMQP protocol version handshake, as described below.

2.3 AMQP Protocol Version Handshake

Prior to sending any frames on a connection, each peer MUST start by sending a protocol header that indicates the protocol version used on the connection. The protocol header consists of the upper case ASCII letters “AMQP” followed by a protocol id of zero, followed by three unsigned bytes representing the major, minor, and revision of the protocol version (currently 1 (MAJOR), 0 (MINOR), 0 (REVISION)). In total this is an 8-octet sequence:

4 octets	1 octet	1 octet	1 octet	1 octet
“AMQP”	%d0	major	minor	revision

This protocol header MUST be sent as the contents of a single WebSocket message.

More details on the version negotiation process is described in Section 2.2 of the AMQP 1.0 Specification **[AMQP]**.

2.4 Exchanging AMQP Frames

After the protocol version handshake is complete, all traffic over the connection is structured as AMQP frames. The first step is to open an AMQP Connection, via the exchange of AMQP open frames. A single WebSocket connection maps to a single AMQP connection. As is normal for AMQP, there MAY be many AMQP sessions over a single WebSocket Connection / AMQP Connection and there MAY be many Links sharing a single Session.

AMQP frames MUST be sent as binary data payloads of WebSocket messages. Specifically, each AMQP frame maps to a single WebSocket message which in turn maps to one or more WebSocket frames. Thus, there is a one-to-many mapping between an AMQP frame and WebSocket frames. For the avoidance of doubt, there is no mapping defined between an AMQP frame and a WebSocket frame. This allows for intermediaries along the communication path to change how a WebSocket message is partitioned in to frames without breaking the AMQP WebSocket binding.

2.4.1 Example

This section is non-normative.

Figure 1 shows an example of how a single AMQP frame maps to a single WebSocket message which is transmitted as two WebSocket data frames.

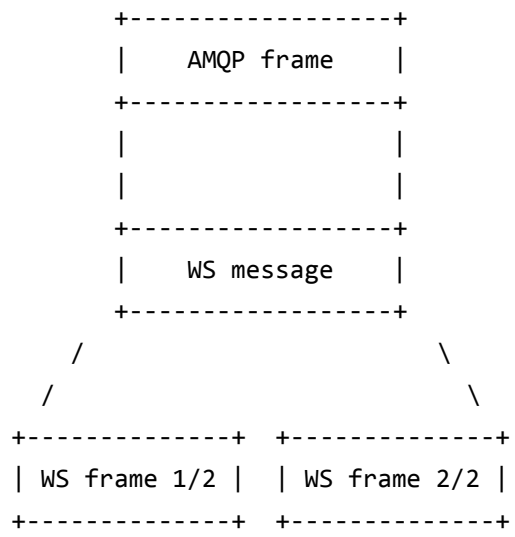

Figure 1: Illustration of how AMQP frames are mapped to WebSocket messages and frames

3 Handling AMQP Redirects

AMQP 1.0 includes two redirect mechanisms, connection-level redirect and link-level redirect. These can be used by one peer to indicate to the corresponding peer that the connection or link should be made to an alternative endpoint.

3.1 Connection-level Redirect

Section 2.8.16 of the AMQP 1.0 specification **[AMQP]** defines the amqp:connection:redirect error which indicates that the container is no longer available on the current connection and that the peer SHOULD attempt reconnection to the container using the details provided in the associated map.

If the amqp:connection:redirect error is received whilst communicating over the WebSocket transport then the peer SHOULD attempt reconnection using the WebSocket transport. That is, cross-protocol redirection is not supported.

The table below indicates how the elements from the info map associated with the amqp:connection:redirect error SHOULD be interpreted when reconnecting:

Redirect error info map element	Usage in the AMQP WebSocket binding
Hostname	This value SHOULD be used in the Host HTTP header of the initial HTTP request. The same value SHOULD be used in the hostname field of the open frame, and during SASL negotiation (if used).
network-host	The DNS hostname or IP address to which the TCP/IP socket SHOULD be opened.
Port	The port number to which the TCP/IP socket SHOULD be opened.

To provide further information for redirection when using the WebSocket transport, the following optional elements MAY also be provided in the info map associated with the amqp:connection:redirect error. The table below indicates how these elements SHOULD be interpreted if provided:

Redirect error info map element	Usage in the AMQP WebSocket binding
Path	This value SHOULD be used as the Request-URI of the HTTP GET request.

3.2 Link-level Redirect

Section 2.8.18 of the AMQP 1.0 specification **[AMQP]** defines the amqp:link:redirect error which indicates that the address provided cannot be resolved to a terminus at the current container.

If the amqp:link:redirect error is received whilst communicating over the WebSocket transport then the peer SHOULD attempt reconnection using the WebSocket transport. That is, cross-protocol redirection is not supported.

The table below indicates how the elements from the info map associated with the amqp:link:redirect error SHOULD be interpreted when reconnecting:

Redirect error info map element	Usage in the AMQP WebSocket binding
Hostname	This value SHOULD be used in the Host HTTP header of the initial HTTP request. The same value SHOULD be used in the hostname field of the open frame, and during SASL negotiation (if used).
network-host	The DNS hostname or IP address to which the TCP/IP socket SHOULD be opened.
Port	The port number to which the TCP/IP socket SHOULD be opened.
Address	The address of the terminus at the container.

To provide further information for redirection when using the WebSocket transport, the following optional elements MAY also be provided in the info map associated with the amqp:link:redirect error. The table below indicates how these elements SHOULD be interpreted if provided:

Redirect error info map element	Usage in the AMQP WebSocket binding
Path	This value SHOULD be used as the Request-URI of the HTTP GET request.

4 Closing a Connection

In the non-error case, the AMQP connection SHOULD be closed first, followed by the WebSocket connection.

Once the AMQP closing handshake has completed, the WebSocket closing handshake should be initiated. As described in **[RFC6455]** section 5.5.1, the peer node desiring to close the connection sends a WebSocket Close frame. Once the other peer node receives this, it MAY finish transmitting any majority finished transmissions, and then MUST send a WebSocket Close frame in return.

5 Security

This binding supports the use of both the standard WebSocket and secure WebSocket protocols. In either case the connection establishment sequence described above remains the same.

For use in standard, non-secure mode, the WebSocket (“ws”) MUST be used as the transport protocol. In this mode, the TCP/IP socket connection SHOULD be made to port 80. For use in secure mode, the WebSocket over SSL or WebSocket Secure (“wss”) MUST be used as the transport protocol. In this mode, the TCP/IP socket connection SHOULD be made to port 443.

Note that the AMQP specification defines two modes of SSL usage, either with a pure SSL tunnel through which the standard AMQP protocol flows, or an in-place upgrade, in which case the transition to SSL occurs after the AMQP protocol handshake. Note that neither of these modes of SSL usage are supported with the AMQP WebSocket binding. If SSL tunneling is required then the WebSocket Secure (“wss”) protocol MUST be used, as described above.

6 IANA Considerations

This specification requests IANA to register the WebSocket AMQP sub-protocol under the “WebSocket Subprotocol Name” registry with the following data:

Subprotocol Identifier	AMQPWSB10
Subprotocol Common Name	WebSocket Transport for Advanced Message Queuing Protocol (AMQP) 1.0
Subprotocol Definition	TBD: URL of this document (when available)

7 Conformance

An implementation is compliant with this specification if it implements all MUST or REQUIRED level requirements.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

- Raphael Cohn, Individual
- Rob Dolin, Microsoft
- Rob Godfrey, JPMorgan Chase Bank, N.A.
- Steve Huston, Individual
- David Ingham, Red Hat
- Alex Kritikos, Software AG, Inc.
- Rafael Schloming, Red Hat
- Jakub Scholz, Deutsche Boerse AG

The following individuals were members of the OASIS Advanced Message Queueing Protocol (AMQP) Binding and Mappings (AMQP-BINDMAP) Technical Committee during the creation of this specification and their contributions are gratefully acknowledged:

- Steve Huston, Individual
- Matthew Arrott, Individual
- Allan Beck, JPMorgan Chase Bank, N.A.
- Andrew Braddock, US Department of Homeland Security
- Raphael Cohn, Individual
- Andrew Doddington, Bank of America
- Rob Dolin, Microsoft
- William Henry, Red Hat
- Ram Jeyaraman, Microsoft
- Alex Kritikos, Software AG, Inc.
- Shawn McAllister, Solace Systems
- Dale Moberg, Axway Software
- John O'Hara, Individual
- Jonathan Poulter, Kaazing
- Oleksandr Rudyy, JPMorgan Chase Bank, N.A.
- Rafael Schloming, Red Hat
- Mark Blair, Credit Suisse
- Laurie Bryson, JPMorgan Chase Bank, N.A.
- Robert Gemmell, JPMorgan Chase Bank, N.A.
- Rob Godfrey, JPMorgan Chase Bank, N.A.
- David Ingham, Microsoft
- Paul Knight, Individual
- Andreas Moravec, Deutsche Boerse AG
- Jakub Scholz, Deutsche Boerse AG
- Wolf Tombe, US Department of Homeland Security

Appendix B. Revision History

Revision	Date	Editor	Changes Made
WD07 (this version)	04 Apr 2014	David Ingham	Fixed typo in Section 2.4.
WD06	03 Apr 2014	David Ingham	Tidied up references and fixed formatting issues in Terminology and References sections.
WD05	02 Apr 2014	David Ingham	Added conformance section.
CSD01 Candidate	25 Feb 2014	David Ingham	Initial version