OASIS Advanced Message Queuing Protocol (AMQP) Addressing Version 1.0

Working Draft

16 April 2013

Specification URIs

This version:

http://docs.oasis-open.org/amqp/addressing/v1.0/wd01/amqp-addressing-v1.0-wd01.xml (Authoritative)
http://docs.oasis-open.org/amqp/addressing/v1.0/wd01/amqp-addressing-v1.0-wd01.html
http://docs.oasis-open.org/amqp/addressing/v1.0/wd01/amqp-addressing-v1.0-wd01.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/amqp/addressing/v1.0/amqp-addressing-v1.0.html
http://docs.oasis-open.org/amqp/addressing/v1.0/amqp-addressing-v1.0.pdf
http://docs.oasis-open.org/amqp/addressing/v1.0/amqp-addressing-v1.0.xml (Authoritative)

Technical Committee:

OASIS Advanced Message Queuing Protocol (AMQP) TC

Chairs:

Robert Godfrey (robert.godfrey@jpmorgan.com), JPMorgan Chase & Co.
Ram Jeyaraman (Ram.Jeyaraman@microsoft.com), Microsoft

Editors:

Robert Godfrey (robert.godfrey@jpmorgan.com), JPMorgan Chase & Co.
David Ingham (David.Ingham@microsoft.com), Microsoft
Rafael Schloming (rafaels@redhat.com), Red Hat

Abstract:

TODO

Status:

This document was last revised or approved by the membership of OASIS 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 `http://www.oasis-open.org/committees/amqp/`.

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 (`http://www.oasis-open.org/committees/amqp/ipr.php`).

Citation format:

When referencing this specification the following citation format should be used:
[amqp-addressing-v1.0]
OASIS Advanced Message Queuing Protocol (AMQP) Addressing Version 1.0. 16 April 2013. OASIS Working Draft.
http://docs.oasis-open.org/amqp/addressing/v1.0/wd01/amqp-addressing-v1.0-wd01.pdf.

Notices:

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 `http://www.oasis-open.org/policies-guidelines/trademark` for above guidance.

Contents

1 Introduction

The Advanced Message Queuing Protocol (AMQP) is an open internet protocol for business messaging. The AMQP Addressing specification defines an addressing syntax and global routing semantics that build on the AMQP Core protocol specification.

1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119].

The authoritative form of the AMQP Addressing specification consists of a set of XML source documents. These documents are transformed into PDF and HTML representations for readability. The machine readable version of the AMQP DTD describes the XML used for the authoritative source documents. This DTD includes the definition of the syntax used in the excerpts of XML presented in the PDF and HTML representations.

1.2 Normative References

[ASCII]
American National Standards Institute, Inc., *American National Standard for Information Systems, Coded Character Sets - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)*, ANSI X3.4-1986, March 26, 1986.

[IANAHTTPPARAMS]
IANA (Internet Assigned Numbers Authority), *Hypertext Transfer Protocol (HTTP) Parameters*.
`http://www.iana.org/assignments/http-parameters/http-parameters.xml`

[IANAPEN]
IANA (Internet Assigned Numbers Authority), *Private Enterprise Numbers*.
`http://www.iana.org/assignments/enterprise-numbers`

[IANASUBTAG]
IANA (Internet Assigned Numbers Authority), *Language Subtag Registry*.
`http://www.iana.org/assignments/language-subtag-registry`

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

[RFC2234]
D. Crocker, Ed., P. Overell, *Augmented BNF for Syntax Specifications: ABNF*. IETF RFC 2234, November 1997.
`http://www.ietf.org/rfc/rfc2234.txt`

[RFC2616]
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*. IETF RFC 2616, June 1999.
`http://www.ietf.org/rfc/rfc2616.txt`

[RFC4122]
P. Leach, M. Mealling, R. Salz, *A Universally Unique IDentifier (UUID) URN Namespace*. IETF RFC 4122, July

2005.
`http://www.ietf.org/rfc/rfc4122.txt`

[RFC4366]
S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, T. Wright, *Transport Layer Security (TLS) Extensions*. IETF RFC 4366, April 2006.
`http://www.ietf.org/rfc/rfc4366.txt`

[RFC5246]
T. Dierks, E. Rescorla., *The Transport Layer Security (TLS) Protocol Version 1.2*. IETF RFC 5246, August 2008.
`http://www.ietf.org/rfc/rfc5246.txt`

[UNICODE6]
The Unicode Consortium. The Unicode Standard, Version 6.0.0, (Mountain View, CA: The Unicode Consortium, 2011. ISBN 978-1-936213-01-6)
`http://www.unicode.org/versions/Unicode6.0.0/`

1.3 Non-normative References

[AMQPCONNCAP]
AMQP Capabilities Registry: Connection Capabilities
`http://www.amqp.org/specification/1.0/connection-capabilities`

[AMQPCONNPROP]
AMQP Capabilities Registry: Connection Properties
`http://www.amqp.org/specification/1.0/connection-properties`

[AMQPFILTERS]
AMQP Capabilities Registry: Filters
`http://www.amqp.org/specification/1.0/filters`

[AMQPLINKCAP]
AMQP Capabilities Registry: Link Capabilities
`http://www.amqp.org/specification/1.0/link-capabilities`

[AMQPLINKPROP]
AMQP Capabilities Registry: Link Properties
`http://www.amqp.org/specification/1.0/link-properties`

[AMQPNODEPROP]
AMQP Capabilities Registry: Node Properties
`http://www.amqp.org/specification/1.0/node-properties`

[AMQPSESSCAP]
AMQP Capabilities Registry: Session Capabilities
`http://www.amqp.org/specification/1.0/session-capabilities`

[AMQPSESSPROP]
AMQP Capabilities Registry: Session Properties
`http://www.amqp.org/specification/1.0/session-properties`

[AMQPSOURCECAP]
AMQP Capabilities Registry: Source Capabilities
`http://www.amqp.org/specification/1.0/source-capabilities`

[AMQPTARGETCAP]
AMQP Capabilities Registry: Target Capabilities
`http://www.amqp.org/specification/1.0/target-capabilities`

1.4 Conformance

TODO

1.5 Acknowledgements

The following individuals have contributed significantly towards the creation of this specification and are gratefully acknowledged:

- David Ingham (Microsoft)
- Gordon Sim (Red Hat)
- Rafael Schloming (Red Hat)
- Rob Dolin (Microsoft)
- Robert Godfrey (JPMorgan Chase & Co.)
- Steve Huston (Individual)

The following individuals were members of the OASIS AMQP Technical Committee during the creation of this specification and their contributions are gratefully acknowledged:

- Alex Kritikos (Software AG, Inc.)
- Allan Beck (JPMorgan Chase Bank, N.A.)
- Andreas Moravec (Deutsche Boerse AG)
- Angus Telfer (INETCO Systems Ltd.)
- Chet Ensign (OASIS)
- Dale Moberg (Axway Software)
- David Ingham (Microsoft)
- Gordon Sim (Red Hat)
- Jakub Scholz (Deutsche Boerse AG)
- James Kirkland (Red Hat)
- John O'Hara (Individual)
- Jonathan Poulter (Kaazing)
- Laurie Bryson (JPMorgan Chase Bank, N.A.)
- Matthew Arrott (Individual)
- Oleksandr Rudyy (JPMorgan Chase Bank, N.A.)
- Rafael Schloming (Red Hat)
- Ram Jeyaraman (Microsoft)
- Raphael Cohn (Individual)
- Rob Dolin (Microsoft)
- Rob Godfrey (JPMorgan Chase Bank, N.A.)
- Robert Gemmell (JPMorgan Chase Bank, N.A.)
- SURYANARAYANAN NAGARAJAN (Software AG, Inc.)

- Sandeep Puri (Cisco Systems)
- Sanjay Aiyagari (VMware, Inc.)
- Steve Huston (Individual)
- William Henry (Red Hat)
- Wolf Tombe (US Department of Homeland Security)

1.6 Revision History

2013-05-29 : Working Draft 1

- Initial Draft

2 Scenarios

2.1 Federation

2.1.1 Configuration

Several brokers from multiple distinct vendors participating in a single federation.

2.1.2 Assumptions

The federation shares a common namespace for addresses. Participating brokers are assigned a distinct domain within the shared namespace.

Each broker has a means to recognize and differentiate between local and non local addresses. Local addresses reference the domain assigned to that broker, and non local addresses reference a non local domain. Furthermore, each broker may have a means to proxy messages and subscription requests to/from non local addresses, thereby functioning as a gateway for some or all of the other brokers participating in the federation. The mechanism for accomplishing and/or configuring this proxying is not specified.

2.1.3 Summary

A standard syntax for differentiating between local and non local portions of an address provides a convenient place to tie in vendor specific configuration mechanisms, thereby encouraging a consistent experience for clients accessing the federated network through gateways provided by distinct vendors.

2.1.4 Examples

- /broker
- /brokerA/queue
- /brokerA/queue/subscription
- /brokerB/topic
- /brokerB/topic/sub-topic
- /brokerB/topic/sub-topic/sub-sub-topic
- /brokerC/service

2.2 Multilevel Federation

2.2.1 Configuration

Messages traveling through a network consisting of multiple federations in different administrative domains.

2.2.2 Assumptions

The multilevel federation shares a common namespace for addresses. Participating federations are are assigned a distinct domain within the shared namespace, and each broker in a federation is assigned a distinct name within its federation.

Each broker has a means to recognize and differentiate between broker local, federation local, and non local addresses. Broker local addresses reference both the federation and the name assigned to that broker, federation local addresses reference the federation, but not the local broker, and non local addresses reference an external federation.

Brokers may have configuration that allows proxying messages and/or subscription requests to/from non broker local addresses, thereby functioning as a gateway for some or all of the other brokers participating in the overall federation. The mechanism for accomplishing and/or configuring this proxying is not specified.

2.2.3 Summary

A standard hierarchical syntax allowing the structure of a multilevel federation to be reflected in the address syntax provides a convenient place to tie in vendor specific configuration mechanisms, thereby encouraging a consistent experience for clients accessing the federated network through gateways provided by distinct vendors.

2.2.4 Examples

- /departmentA
- /departmentA/broker
- /departmentA/broker1/queue
- /departmentA/broker1/queue/subscription
- /departmentA/broker2/topic
- /departmentA/broker2/topic/sub-topic
- /departmentA/broker2/topic/sub-topic/sub-sub-topic
- /departmentA/broker3/service
- /departmentB/broker
- /departmentB/broker1/queue
- /departmentB/broker1/queue/subscription
- /departmentB/broker2/topic
- /departmentB/broker2/topic/sub-topic
- /departmentB/broker2/topic/sub-topic/sub-sub-topic
- /departmentb/broker3/service

2.3 DNS

2.3.1 Configuration

A large scale loosely coupled federation of brokers, AMQP services, and AMQP-aware transparent intermediaries.

2.3.2 Assumptions

The number of participating elements in this federation is large enough to make it impractical to manually configure all routes between participants.

Participants in this federation leverage kerberos, SSL certs, claims based auth, or some other authentication scheme such that pairwise configuration of each participant is not necessary in order to acheive mutual authentication.

2.3.3 Summary

For such a deployment it is valuable to be able to leverage existing DNS infrastructure for both provisioning namespaces within the federation, and for establishing routes between the more loosely coupled portions of the federation. The addressing syntax can enable this kind of deployment by providing a clear way to identify when the root name is actually a DNS name that can be used to look up an SRV, A, or AAAA record for use in establishing a route.

2.3.4 Examples

- //example.com
- //example.com/queue
- //example.com/queue/subscription
- //example.com/topic
- //example.com/topic/sub-topic
- //example.com/topic/sub-topic/sub-sub-topic
- //example.com/service
- //amqp.example.com
- //amqp.example.com/queue
- //subdomain.amqp.example.com
- //subdomain.amqp.example.com/queue
- //example.com/departmentA/broker2/queue

2.4 Single Broker

2.4.1 Configuration

A single broker that is only aware of its own namespace.

2.4.2 Assumptions

Such a broker may directly contain entities with simple names such as queues, topics, or other nodes, and may also define their own syntax to expose structure within their own namespace.

2.4.3 Summary

Brokers implemented and deployed before this addressing standard my not recognize the address syntax or semantics defined herein. It is therefore desirable to minimize the chance that non local addresses will conflict with a single broker using addresses to directly reference entities within its own namespace.

2.4.4 Examples

- queue
- queue/subscription
- topic
- topic/sub-topic
- topic/sub-topic/sub-sub-topic
- topic.sub-topic
- topic.sub-topic.sub-sub-topic
- topic:sub-topic:sub-sub-topic
- topic::sub-topic::sub-sub-topic
- topic://name
- queue://name
- service

2.5 Request/Response

2.5.1 Configuration

A request/response style service built on top of the AMQP protocol.

2.5.2 Assumptions

Clients may access the request/response service through a variety of topological configurations, including:

- A direct connection with no intervening intermediaries.
- A single traditional broker.
- A federation of brokers.
- One or more transparent intermediaries

The service itself does not act as a general purpose broker. It is not capable of creating temporary queues.

2.5.3 Summary

A request/response scenario would benefit greatly from a way to route responses that does not depend on the topology used to access the service. It is undesirable to force the use of an intermediary to host a temporary queue, and likewise undesirable to force the choice of a specific location for the temporary queue in a federated scenario. As such it is useful for the endpoints themselves to participate directly as addressible entities in the overal network topology.

2.5.4 Examples

- /service
- /broker/service
- //example.com/service
- /client-1234
- /client-4321/request-27

3 Requirements

- AMQP addresses support global transcription as defined by RFC3986 section 1.2.1.
- AMQP addresses may optionally advertise accessability via a DNS based authority for scenarios where an AMQP endpoint has well known DNS records.
- AMQP addresses may optionally advertise accessability via a literal IP/port address for scenarios where an AMQP endpoint has no DNS records. This literal syntax must include the same information available via SRV records including IP, port, and the protocol stack in use, e.g. tls over tcp vs dtls over sctp, vs web sockets.
- AMQP addresses may also be used to uniquely identify global endpoints that do not have any associated DNS records nor a stable IP address.
- The AMQP addressing specification defines the relationship between the *to* field and the *source/target* fields such that transparent intermediaries can operate at the message level. This means that messages created by implementations conforming to this specification may be multiplexed from N links down to a single link without losing any information about the ultimate destination of the message. Furthermore it should be possible to demux from a single link back to N links bases solely upon the contents of the *to* field.
- The AMQP addressing specification defines how link targets may be used to establish dynamic routes to/from a remote source. In particular, this enables a dynamic request/response pattern that does not require creating any intermediate temporary nodes.

4 Addresses

4.1 Syntax

The AMQP addressing standard defines a path syntax for describing globally routable AMQP addresses: Unresolved elements below are defined by RFC3986:

```
    address = "/" domain [ "/" path ]

     domain = dns-domain
            / amqp-domain

 dns-domain = "/" reg-name
            / "/" srv-literal

srv-literal = [ proto ":" ] host [ ":" port ]
      proto = [a-zA-Z_]+
       port = *DIGIT
       host = IP-literal / IPv4address

amqp-domain = name

       path = name [ "/" path ]
       name = [^/]*
```

Figure 4.1: BNF

4.2 Semantics

An AMQP global address has two parts: a domain and a path. The path MAY be omitted, however a domain MUST be present to form a valid global AMQP address. The domain is either an amqp domain or a dns domain. An amqp domain consists of a simple name. A dns domain contains either a registered name or a literal representation of the service record info that is obtained from a registered name.

4.3 DNS Domain Resolution

A DNS domain MAY be resolved to a protocol, an ip address, and port number. The currently defined protocols are “tcp” and “tls”. (TODO: define what each protocol means) The default protocol is defined to be “tcp”, and the default port number is 5672. (TODO: should we define a protocol and/or port number for either?)

A DNS domain consists of either a registered name or an srv-literal. A registered name is resolved by first querying any SRV records for that name. If no service records exist, then any A or AAAA records are used to obtain an ip address, and the default protocol and port are assumed.

If the DNS domain is an srv-literal, the protocol, ip address, and port are taken directly from their literal form inside the domain portion of the address.

4.4 Patterns

This specification defines a simple pattern matching syntax for describing classes of addresses.

4.4.1 Address Pattern

```
<type name="address-pattern" class="restricted" source="string">
    <descriptor name="amqp:address-pattern:string" code="0x00000000:0x000000??"/>
</type>
```

Every valid address is also a pattern that matches the address exactly. Additionally a pattern may include the wildcard characters ’%’ and ’*’ which are not allowed in valid AMQP addresses. The ’%’ wildcard matches any valid address character with the exception of ’/’. The ’*’ wildcard matches all valid address characters with no exceptions. TODO: define a real descriptor here

5 Routing

5.1 Routable Messages

A routable message MUST include the entire valid AMQP address in the to field of the message. A routable message MUST reach its destination when traveling over a link with a target address that is identical to the *to* address. If the target address is not identical to the *to* address, then the behavior is defined by the target node.

5.2 Anonymous Terminus

A source or target with a *null* address is refered to as an *anonymous terminus* and is defined to resolve to a special *relay node*. A relay node acts as a proxy for links to other nodes. Messages sent over links into a relay node will be relayed to the node referenced in the to field of the message just as if a direct link has been established to that node. Links coming out of a relay node will relay messages from all nodes identified by the address pattern(s) in the source node filter, just as if a direct links have been established from those nodes. (TODO: are we defining the null address once and for all or should we have a special property to indicate that we are actually talking to a relay node?)

5.3 Legacy Addresses

Address strings appearing in protocol fields with a value that does not start with "/" are considered legacy addresses. These are assumed to be scoped to a single container. There is no way to know in general to which container they are scoped. Implementations MAY assume that these addresses are scoped to the container that produces them.

5.4 Address Equivalence

Containers participating in non local namespaces that also support legacy addresses MUST recognize the following forms as equivalent where <domain> represents the name of the container in the non local namespace.:

- /<domain>/<path>
- <path>

Containers MAY participate in multiple non local namespaces and as such recognize even more equivalent forms for a given address.

5.5 Dynamic Routes

A container MAY advertise the existence of a node accessible via a given address by placing that address in the source or target of a link to a remote node. A remote node supporting dynamic routing MUST relay incoming messages with a matching *to* address down an authorized incoming link with a matching target.

A remote container supporting dynamic routing MUST allow authorized links to/from the advertised address and relay any messages to/from the advertising link(s).

6 TODO

- define properties/capabalities
- define SRV records