AMQP Addressing Version 1.0

Committee Specification Draft 01

17 March 2021

This stage:
https://docs.oasis-open.org/amqp/addressing/v1.0/csd01/addressing-v1.0-csd01.docx (Authoritative)
https://docs.oasis-open.org/amqp/addressing/v1.0/csd01/addressing-v1.0-csd01.html
https://docs.oasis-open.org/amqp/addressing/v1.0/csd01/addressing-v1.0-csd01.pdf

Previous stage of Version 1.0:
N/A

Latest stage of Version 1.0:
https://docs.oasis-open.org/amqp/addressing/v1.0/addressing-v1.0.docx (Authoritative)
https://docs.oasis-open.org/amqp/addressing/v1.0/addressing-v1.0.html
https://docs.oasis-open.org/amqp/addressing/v1.0/addressing-v1.0.pdf

Technical Committee:
OASIS Advanced Message Queuing Protocol (AMQP) TC

Chairs:
Rob Godfrey (rgodfrey@redhat.com), Red Hat
Clemens Vasters (clemensv@microsoft.com), Microsoft

Editor:
Clemens Vasters (clemensv@microsoft.com), Microsoft

Related work:
This document 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 Addressing specification further defines the “AMQP network” concept introduced in the main AMQP specification as a federation of AMQP containers whose nodes communicate with each other either directly or via intermediaries. This specification also defines the semantics of the “address” archetype that was left undefined in the main AMQP specification, and the syntax for the AMQP URI scheme and a matching restriction of the AMQP “address-string” type.

Status:
This document was last revised or approved by the OASIS Advanced Message Queuing Protocol (AMQP) TC on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=amqp#technical.

TC members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/amqp/.

This specification is provided under the RF on RAND Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. 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 TC's web page (https://www.oasis-open.org/committees/amqp/ipr.php).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Key words:
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

Citation format:
When referencing this document, the following citation format should be used:

[Addressing-v1.0]

AMQP Addressing Version 1.0. Edited by Clemens Vasters. 17 March 2021. OASIS Committee Specification Draft 01. https://docs.oasis-open.org/amqp/addressing/v1.0/csd01/addressing-v1.0-csd01.html. Latest stage: https://docs.oasis-open.org/amqp/addressing/v1.0/addressing-v1.0.html.

Notices:

Table of Contents

1 Introduction

The core AMQP specification [AMQP 1.0] introduces the concept of an *AMQP network* as the conceptual foundation for its architectural elements:

> *An AMQP network consists of nodes connected via links. Nodes are named entities responsible for the safe storage and/or delivery of messages. Messages can originate from, terminate at, or be relayed by nodes.*
>
> *[…]*
>
> *Nodes exist within a container. Examples of containers are brokers and client applications. Each container MAY hold many nodes. Examples of AMQP nodes are producers, consumers, and queues.*

While the AMQP network concept is referenced several times within the core specification, there is no formal definition of the network model in the core specification since it is primarily focused on defining a peer-to-peer transfer model and protocol.

This specification provides an expanded conceptual framework for AMQP networks and for addressing the elements within them. It also formally defines the schema and syntax of the AMQP Uniform Resource Identifier.

The “AMQP Networks” section provides the conceptual framework, including examples of its application. “Addressing Elements” defines the elements of the addressing model, which includes formal constraints on the use of addressing-related constructs in the core AMQP specification.

The “AMQP address” section defines the Uniform Resource Identifier syntax.

1.1 Glossary

1.1.1 Definitions of terms

When used in this specification and unless explicitly stated otherwise, the term “message” always refers to an AMQP message using the default message format of [AMQP 1.0, 3.2.16].

2 AMQP Networks

The core AMQP specification defines an *AMQP network* to consist of *nodes* connected via *links*, and nodes existing within *containers*. The objective of the AMQP core specification is to define a peer-to-peer protocol for transferring messages between nodes in an AMQP network, and a protocol for establishing communication between nodes residing in containers.

The AMQP concept of “network” is more abstract than that of the Internet Protocol (IP) and similar networking protocols. An AMQP network has no firm associations with underlying resources in the way that DNS names map to IP addresses and through to network hardware identifiers.

AMQP’s nodes and *containers* are application concepts and therefore, resolving address information (or message metadata) to AMQP network locations, and routing messages towards those locations are application-level tasks.

As a result, an AMQP network allows an application to define an overlay network routing model that spans one or more underlying transport networks and multiple middleware infrastructures.

Safety and security critical environments often use isolated IP networks that are only reachable through application-layer gateways acting as an “air gap” for IP traffic. IP-level access to the protected equipment potentially opens up a large attack surface area, and unauthorized access and manipulations may result in safety hazards that could result in destroyed equipment, injury, or death.

The network segmentation and mutual isolation employed in such scenarios makes it intentionally impossible to establish an IP route to services or equipment in the protected networks.

The AMQP network model aims to make it possible for AMQP intermediaries in gateways and devices to securely route application-level messages from (authorized) external parties across the boundaries of isolated IP networks without exposing unsecured or vulnerable network assets, and while enabling the information flow to be inspected by intermediaries and logged in audit trails.

When supported by the container implementations, the origins of routed AMQP traffic can also be masqueraded [MESSAGE-ANNOTATIONS-V1.0] similar to how Network Address Translation (NAT) performs it for IP, protecting sender privacy and preventing discovery of routing topologies by receivers not privileged to such information. A robot manufacturer might be authorized to see information about a robot, but it’s not necessarily authorized to discover the entire factory routing topology.

The integration of an AMQP network with an IP network is performed using “on-ramp” endpoints. An “on-ramp” is an IP-addressable container that is part of an AMQP network and to which external clients can connect. Those external clients (technically AMQP containers themselves) then establish links and transfer messages to and through the “on-ramp” container.

While AMQP containers are typically interconnected via IP networks, the IP routes between containers do not play a role in AMQP addressing. An AMQP container, like one that’s hosted inside a moving vehicle or ship or aircraft, might quite well be changing IP networks and addresses and therefore inter-container routing, but its AMQP address will remain stable.

2.1 Nodes

Nodes are the sources or targets of messages in an AMQP network. AMQP does not differentiate between clients and servers, but only knows communicating peers. Implementations using AMQP might assign specific roles to nodes, like producer and queue and consumer, but these differences play no role at the AMQP level. A producer sending messages to a queue is just an AMQP node transferring messages to another node.

AMQP *links* exist to establish transfer routes between *nodes,* whereby the nodes on either end may reside in different or the same the same container or may be distributed across different containers.

Nodes typically reflect distinct architectural entities in the implementing container that are accepting or emitting messages. The AMQP container resolves addresses to internal objects.

2.2 Containers

AMQP containers are naming and management scopes for AMQP nodes.

A container might be a single process on a single machine, but it might also be a distributed system spanning many processes and machines and using private means to present the container as one towards external parties.

AMQP *connections* and *sessions* exist to establish communication paths for AMQP *links* across an underlying transport network and between containers.

The core AMQP specification defines a binding of the AMQP connection layer to TCP, optionally with overlaid TLS. The AMQP Web Socket Binding specification defines a binding to the Web Socket protocol, which allows for network endpoint sharing with HTTPS.

An AMQP container MAY be concurrently reachable via multiple, different underlying transport network endpoints and using different protocol bindings. Especially, an AMQP container MAY be reachable via different transport network endpoints that are each bound to networks which are otherwise unrelated.

This specification mandates that each container MUST use an AMQP network-unique *container-id* to identify itself during connection establishment in the open performative.

2.3 Scopes

AMQP *nodes* and *containers* are introduced in the core AMQP specification. This specification introduces a further addressing concept: Scopes.

A scope is an addressing abstraction at the level of AMQP containers that either describes where a message or link originates from, or where a message or link is destined to go. Scopes MAY form hierarchies and the MAY map to one or multiple containers.

A *scope identifier* is a structured name that represents a logical routing target or source, meant to be used for identifying destinations for messages or links across AMQP container boundaries, potentially with one or more containers and their nodes acting as routing intermediaries. The term *scope* reflects that the addressed realm has further internal structure (nodes).

Scope identifiers follow DNS naming conventions, but they do not have to be registered in DNS and they don't have an associated IP address. Even if scopes are not registered in DNS, ownership rights of corresponding DNS domain name owners should be respected. Scope names are just names describing logical realms in a system, and the DNS-like structure helps expressing hierarchical relationships.

For example, scope identifiers MAY be any of the following:

- Arbitrary names for use in a private network with its own naming conventions
- UUID-based names
- Subdomain-names of a registered DNS domain to provide internet-unique naming. DNS scope names do not need to have any IP addresses registered.

Scope expressions are matching conditions that can be evaluated against *scope identifiers* for routing or filtering. Instead of having to handle container-ids for all potential routing targets, routing intermediaries can evaluate *scope expressions* to determine which container to direct the message to.

2.4 Relationship between Scope and Container Identifiers

While the identifier of a container MUST be unique and stable within an AMQP network, scopes and containers have a many-to-many relationship. Many containers can belong to one scope, a single container can belong to many scopes.

If a message is sent to an address with a scope, then the AMQP network MUST deliver that message to a container in that scope. If an address does not have a scope, then it MAY be handled by immediately connected container or MAY be forwarded to any other container.

How an AMQP network maps scope names to container names, or determines membership of a scope, is not defined by this specification.

In the core AMQP specification, the container identifier (container-id) is used for containers to identify themselves to the communicating peer in the OPEN performative that is exchanged during connection establishment, but it is never used to locate the container.

The *scope identifier* introduced in this specification is meant to facilitate locating containers, but it does not define or constrain specific methods by which the container is located.

Non-normative example:

Consider a lookup table inside of a container acting as a sender that pairs *scope expressions* with AMQP URIs, each identifying a peer container handling further routing. The “Target URI” column uses the AMQP URI format defined in section 4.

Scope expression	**Target URI**
singapore.southeast-asia.amqp.org	amqps://sea-2.example.com/
***.southeast-asia.amqp.org**	amqps://sea-1.example.com/queue1
west-europe.amqp.org	amqps://weu-1.example.com/(europe.amqp.org)/
***.amqp.org**	amqp:(world.example.com)

When the sending container needs to resolve a scope identifier, for example by encountering it in an AMQP URI either in a link target address or inside the *to* property of a message, it consults the lookup table for finding a *scope expression* match, and the lookup yields a target URI if a match is found.

The target URI is then used to determine the next routing activities. Those might involve connecting to the network endpoint indicated in the URI or finding some private configuration information for creating such a connection. Scope and path information in the target URI might be used to further direct the message or link flow. The specifics of how that might occur are not defined in this specification.

Whether the identified target container is the ultimate destination, or whether its own configuration instructs it to route links or messages further onwards is a private concern of the target container. The sending container therefore MUST NOT make any assumptions about how the *container-id* received via the target container’s OPEN performative relates to the *scope identifier*. The only relevance of the *container-id* is that for link recovery, the *container-id* of a newly established recovery connection MUST match that of the prior connection, and the *container-id* might also be handy as a lookup key for established connections.

When the *scope identifier* is absent from a fully qualified address expression or when it is empty, the address targets the “current” container, whereby the “current” container is where the address expression is being evaluated. For instance, for a link sending messages from container A to container B, the source address is considered relative to A, and the target address is relative to B.

3 Addressing Elements

AMQP addresses are used for multiple purposes:

- establish IP network connections between containers
- establish links between nodes within the same container or across different containers
- route messages inside or across containers based on their addressing metadata

To help with these tasks, an address needs to be able to hold the following information:

1. A protocol scheme to indicate how communication should be initiated when a network connection is required.
2. An “authority” that includes network endpoint information for establishing a connection.
3. A scope identifier that indicates which logical AMQP network scope the message or link is directed towards.
4. A path expression that maps to a node (or a node terminus) inside the target container.
5. A set of application-defined parameters that allow embedding information in the URI that is required for establishing the desired communication path.

3.1 Protocol Schemes

For the core AMQP transport, the defined schemes are *amqp* for regular AMQP connections with in-band TLS upgrades, and *amqps* for the alternate establishment connection mode that begins with TLS connection (Section 5.2.1) [AMQP 1.0].

The default TCP port for the amqp scheme is 5672. The default TCP port for the amqps scheme is 5671.

AMQP WebSocket Binding [AMQP-WS] endpoints MUST be described with the standard *ws* (non-secure) or *wss* (*secure, TLS*) WebSocket schemes.

The default TCP port for the *ws* scheme is 80. The default TCP port for the *wss* scheme is 443.

For WebSockets, the client will subsequently rely on the *amqp* WebSocket subprotocol negotiation for discovering whether the endpoint does indeed support AMQP 1.0.

The *scope* scheme is a network- and transport-protocol independent scheme for URIs that only carry *scope* and *path* information. While this specification defines the *scope* scheme and its syntax, *scope* URIs and the supporting abstract address model can also be used with overlay networks that are not AMQP based or that use a mix of AMQP and other protocols.

3.2 Network Endpoint

While AMQP’s addressing model is primarily a high-level abstraction for overlay networks, external clients outside of such an overlay network must be able to find an entry point (“on-ramp”) into such an overlay network, and the overlay network parties need to be able to locate each other on the underlying network as well. Therefore, some address expressions need to carry both logical addressing information as well as network endpoint addressing information.

For the current AMQP transport bindings, the network endpoint will generally be an IP network address or an IP address resolvable hostname along with a TCP port number. The network endpoint’s port number is only required if it deviates from the protocol scheme’s default. For uses of AMQP with non-IP transports, the conformance rules spelled out here are equivalently applicable.

The IP network endpoint identifies the “on-ramp” into an AMQP network; it helps an otherwise external party (typically referred to as “client”) to establish a connection to a container that is part of said AMQP network.

The network endpoint information is formally called “authority” in the URI syntax [RFC3986], but for clarity this specification refers to it as “network endpoint”.

3.2.1 Link Addresses

Link addresses (as used in *target* address and *source* address of the “attach” performative) SHOULD NOT include a network endpoint.

Links are made over a connection which has already been established, so a network endpoint is redundant. If a link address has a network endpoint this is not an error, but it MUST be ignored (this allows AMQP addresses to be copied without checking for a network endpoint).

3.2.2 Message 'to' field

The 'to' field of a message MAY contain a network endpoint.

Note the 'to' field is part of the bare message which may be signed and cannot be modified by intermediaries. Any message recipient MUST ignore the network endpoint and not use it as a dispatch criterion, as access via different on-ramps to the same AMQP address is equivalent.

An intermediary who forwards messages MAY ignore the 'to' field and forward a message within its own network, or MAY connect to the 'to' field network address. How such intermediaries make this decision is out of scope of this specification; they MAY use custom annotations, properties of the link or connection that received the message, or other mechanisms.

3.2.3 Message 'reply-to' and the Request Reply pattern

The 'reply-to' field MAY contain a network endpoint.

Message recipients and intermediaries SHOULD prioritize attempting to deliver replies over the same connection that through which the message was obtained. If this is impossible or fails, the response SHOULD be sent via an outbound connection to the 'reply-to' network endpoint.

3.3 Scope Identifier

As discussed in 2.3 the scope identifier is used to direct messages to the appropriate container.

If the scope identifier for a terminus is empty, it is interpreted as to belong to the scope of the terminus. For instance, when attaching a link to receive messages, if the source address’ scope identifier is empty, the implied scope is that of the container that contains the source from which messages are being sent.

An AMQP address consisting of a network endpoint and an empty scope identifier will result in the overall address expression being resolved by the container reachable via the network endpoint or its equivalent communication path.

An AMQP address consisting of a network endpoint and a non-empty scope identifier will result in the overall address expression resolving to a container identified by the scope identifier while the network endpoint or its equivalent communication path serve as “on-ramp” into the AMQP network if required.

How a scope identifier maps to target containers and how it is determined that a message or link has reached its intended destination is outside the scope of this specification and implementation specific.

3.4 Path

As discussed in 2.1, the path expression is a sequence of identifier segments that reflects a path through an implementation specific relationship graph of AMQP nodes and their termini. The path expression MUST resolve to a node’s terminus in an AMQP container. An empty path expression reflects the anonymous terminus.

How the path expression relates to the graph and terminus is outside the scope of this specification and implementation specific.

3.5 Parameters

Parameters are a set of application-specific key-value pairs carried alongside the other address information. The OASIS AMQP TC reserves parameter name prefixed with “amqp:” for use in its specifications.

Parameters are useful when extra information needs to be given to the party handling the address information.

For instance, when a sending application wants to pass a *reply-to* address to a message consumer and the sender wants to grant the consumer limited access to the reply destination, it might include a parameter that carries a security token granting the required access.

```
amqp://endpoint.example.com/(site-b.contoso.com)/queue?sec-token=…
```

4 The AMQP Address

An *AMQP address* is a *URI reference* as defined by RFC3986.

This specification defines schemes “amqp” for plain TCP connections and “amqps” for TLS connections.

Implementations MAY support other connection schemes (for example ws, wss) and other AMQP specifications MAY introduce other AMQP-specific schemes.

4.1 Transport independent addresses

A *transport-independent address* has no network information and cannot be used to connect to an AMQP network endpoint, but it can be used to send messages once connected – as a link source or destination address, or a message “to” or “reply-to” address.

This is compatible with AMQP practice prior to this specification. In terms of RFC3986, such an address is a *relative URI reference*. For example:

- myqueue
- amqp:myqueue
- /area/mailbox
- /(site.example.com)/foo/bar/thing

Note in the last example “site.example.com” is an AMQP scope identifier, not a DNS name.

4.2 AMQP URLs

A URL provides a network endpoint to connect to an AMQP service, and the address of an AMQP node, for example:

- amqps://onramp.example.com/(site.net)/target

An AMQP URL may have no path, for example

- amqp://service.org

Such a URL can be used to establish a connection. If used as the source or target address of a link it refers to the *anonymous terminus* [ANONTERM-V1.0].

4.3 AMQP URI Syntax

The following ABNF notation builds on the defined elements from Appendix A of RFC3986.

New syntax is marked “AMQP Specific”, the rest is directly from RFC3986. This syntax can be parsed using a standard URI parser by examining the first element of the URI path to see if it matches the 'path-amqp-scope' syntax below.

```
URI           = scheme ":" hier-part [ "?" query ] [ "#" fragment ]

scheme        = “amqp” / “amqps” ; AMQP specific

hier-part     = "//" authority path-abempty
              / path-absolute
              / path-rootless
              / path-empty
              / path-amqp-scope  ;; AMQP-specific
```

```
   URI-reference = URI / relative-ref

   absolute-URI  = scheme ":" hier-part [ "?" query ]

   relative-ref  = relative-part [ "?" query ] [ "#" fragment ]

   relative-part = "//" authority path-abempty
                 / path-absolute
                 / path-noscheme
                 / path-empty
                 / path-amqp-scope  ;; AMQP-specific

## AMQP specific:

   path-amqp-scope = "/" amqp-scope path-absolute
   amqp-scope      = "(" reg-name ")"
```

The syntax elements map to the AMQP address elements as follows:

```
      scheme     ::= Protocol Scheme (3.1)
      authority  ::= Network Endpoint (3.2)
      amqp-scope ::= Scope (3.3)
      path-*     ::= Path (3.4)
      query      ::= Parameters (3.5)
```

4.4 Examples

Examples of URLs with network endpoint information.

1. AMQP URLs with just a network endpoint. Those URIs identify the container reachable via the network endpoint and the anonymous terminus of that container:
    - amqp://endpoint.example.com
    - amqp://endpoint.example.com:15671
2. AMQP URLs with a network endpoint and a path. Those URIs identify the container reachable via the network endpoint and the node identified by the path:
    - amqp://endpoint.example.com/queue
    - amqp://endpoint.example.com/area/queue
3. AMQP URLs with just a network endpoint and a parameter. Those URIs identify the container reachable via the network endpoint and the anonymous terminus of that container:
    - amqp://endpoint.example.com:15671/?access_token={token}
4. AMQP URLs with a network endpoint and scope identifier. Those URIs identify a container and its anonymous terminus by the scope identifier and provide an on-ramp endpoint.
    - amqp://endpoint.example.com/(site-a.contoso.com)/
5. AMQP URLs with a network endpoint, scope identifier, and path. Those URIs identify a container by the scope identifier and a node inside the container with an on-ramp endpoint.
    - amqp://endpoint.example.com/(site-b.contoso.com)/queue
    - amqp://endpoint.example.com/(site-c.contoso.com)/area/mailbox

Examples of URIs without network endpoint information. May be used when a connection is already established.

6. AMQP URIs with a scope identifier and path. Those URIs identify a container by the scope identifier and a node inside the container.

- amqp:(site-c.contoso.com)/area/mailbox
- amqp:(site-b.contoso.com)/queue

7. AMQP URIs with just a path. Those URIs identify the current container and a node inside the container.
    - amqp:/queue
    - amqp:/area/mailbox
    - amqp:queue

Examples of URI references without a scheme. May be used where AMQP is implied, for example in AMQP message to/reply-to fields and link source/target address fields. May also be used externally where an application knows that AMQP is intended and has some other way to make a connection.

8. AMQP URI references with a scope identifier and path.
    - (site-c.contoso.com)/area/mailbox
    - (site-b.contoso.com)/queue
9. AMQP URIs with just a path. Those URIs identify the current container and a node inside the container.
    - /queue
    - /area/mailbox
    - queue

5 Conformance

Implementations of this specification are conformant if they can correctly produce and parse AMQP address expressions (URI references) as defined in section 4.3 and all defined AMQP addressing elements from section 3 can be set or accessed on the address expression by an application or an AMQP container.

Appendix A. References

This appendix contains the normative and informative references that are used in this document. Normative references are specific (identified by date of publication and/or edition number or Version number) and Informative references may be either specific or non-specific.

While any hyperlinks included in this appendix were valid at the time of publication, OASIS cannot guarantee their long-term validity.

A.1 Normative References

The following documents are referenced in such a way that some or all of their content constitutes requirements of this document.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3986]

Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5646]

Phillips, A., Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[RFC8174]

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <http://www.rfc-editor.org/info/rfc8174>.

[AMQP 1.0]

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

[MESSAGE-ANNOTATIONS-V1.0]

Message Annotations for Response Routing Version 1.0. Edited by Rob Godfrey. 16 February 2021. OASIS Committee Specification 01. https://docs.oasis-open.org/amqp/respann/v1.0/cs01/respann-v1.0-cs01.html. Latest stage: https://docs.oasis-open.org/amqp/respann/v1.0/respann-v1.0.html.

[AMQP-WS]

OASIS Advanced Message Queuing Protocol (AMQP) WebSocket Binding (WSB) Version 1.0, Edited by John Fallows, David Ingham, and Robert Godfrey. OASIS Standard. http://docs.oasis-open.org/amqp-bindmap/amqp-wsb/v1.0/amqp-wsb-v1.0.html

[ANONTERM-V1.0]

Using the AMQP Anonymous Terminus for Message Routing Version 1.0, Edited by Robert Godfrey. OASIS Committee Specification. http://docs.oasis-open.org/amqp/anonterm/v1.0/anonterm-v1.0.html

Appendix B. Security and Privacy Considerations

The addressing specification builds on the RFC3986 URI/URL format, which permits credentials to be embedded within the authority portion of the URL.

If provided, AMQP implementations MAY use the embedded information from and extract the credentials from the URI, for instance for SASL PLAIN authentication.

Generally, applications SHOULD NOT embed credentials or other secrets in URLs either in the authority section or in query parameters or elsewhere, because URLs are often logged and credentials might therefore leak through logs.

Appendix C. Acknowledgments

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

Alan Conway, Red Hat
Rob Godfrey, Red Hat
Keith Wall, Red Hat
Robbie Gemmell, Red Hat
Justin Ross, Red Hat
Ted Ross, Red Hat
Oleksandr Rudyy, JP Morgan
Xin Chen, Microsoft
Clemens Vasters, Microsoft

Appendix D. 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: [https://www.oasis-open.org/policies-guidelines/ipr].

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 AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

[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 Standards Final Deliverable, 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 deliverable.]

[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 OASIS Standards Final Deliverable 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 OASIS Standards Final Deliverable. 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 OASIS Standards Final Deliverable 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 Standards Final Deliverable, 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 document, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, documents, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.