XEP-0079: Advanced Message Processing

Matthew Miller
mailto:linuxwolf@outer-planes.net
xmpp:linuxwolf@outer-planes.net

Peter Saint-Andre
mailto:stpeter@stpeter.im
xmpp:stpeter@jabber.org
https://stpeter.im/

2005-11-30
Version 1.2

<table>
<thead>
<tr>
<th>Status</th>
<th>Type</th>
<th>Short Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>Standards Track</td>
<td>amp</td>
</tr>
</tbody>
</table>

This specification defines an XMPP protocol extension that enables entities to request, and servers to perform, advanced processing of XMPP message stanzas, including reliable data transport, time-sensitive delivery, and expiration of transient messages.
Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the XMPP Standards Foundation (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

## NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. ##

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF’s Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).
6 Error Handling 18
   6.1 Conditions .............................................. 18
   6.2 Examples .............................................. 20
      6.2.1 Unsupported Action ............................... 20
      6.2.2 Unsupported Condition ......................... 21
      6.2.3 Not Acceptable .................................. 22
      6.2.4 Service Unavailable ............................. 22
      6.2.5 Undefined Condition ............................. 23

7 Implementation Notes 24

8 Stream Feature 24

9 Security Considerations 25

10 IANA Considerations 25

11 XMPP Registrar Considerations 26
   11.1 Protocol Namespaces ................................. 26
   11.2 Stream Features ..................................... 26
   11.3 Well-Known Service Discovery Nodes .............. 26
   11.4 Registries ........................................... 26
      11.4.1 Rule Conditions Registry ..................... 26
      11.4.2 Rule Actions Registry ......................... 28

12 XML Schemas 29
   12.1 AMP .................................................. 29
   12.2 Errors ............................................... 30
   12.3 Stream Feature ...................................... 31

13 Acknowledgements 32
1 Introduction

This document defines a protocol that enables an end-point entity to specify additional delivery semantics for an XMPP <message/> stanza. This protocol is typically used by clients to inform the receiving server how to deliver a particular stanza, such as providing an expiration time or a resource-matching strategy.

1.1 Motivation

The built-in delivery semantics for <message/> stanzas (defined in XMPP Core ¹ and, for instant messaging applications, also in XMPP IM ²) are adequate for most current applications. However, there are various cases where more stringent delivery semantics are necessary. The most common cases discussed in this document are:

- Reliable data transport -- the sender requires notification (positive and/or negative) of message delivery.
- Time-sensitive messages -- the message is valid only until a certain date and time.
- Transient messages -- the message should not be stored offline for later delivery.

1.2 Concepts

This protocol is mostly handled by the server or servers processing the <message/>. The protocol consists of a list of rules, with conditions and actions for each rule. Upon receipt of an appropriately marked message, the server interprets the rules in the order they are received, looking for met conditions. When a condition is met, the action for that rule is executed, and message processing stops.

Each rule is flagged for the scope it applies to, whether it be the overall route, or for each hop in the route. Additionally, while this document defines a default set of conditions and actions, the protocol is extensible enough to allow for more to be defined in the future.

The namespace for the protocol is "http://jabber.org/protocol/amp".

1.3 Prerequisites

In order for this protocol to function properly, the containing <message/> stanza MUST possess an 'id' attribute, and the value of the 'id' attribute MUST NOT be an empty string. The server MUST include this 'id' attribute value with any response back to the sender; this enables servers and clients to properly relate the initial request with any subsequent alert.

error, or notification.

2 Process Flows

2.1 Sender-to-Server

The following use case flow describes the interaction between the sender and a server. As illustrated below, this interaction is actually rather simple:

1. Sender determines support (E1)
2. Sender specifies appropriate rules and sends message to server (E1,E2)
3. Sender awaits any protocol-specific responses (UCE)
   • E1: The server does not support this protocol (UCE unsuccessfully)
   • E2: The server does not support one or more specified rule conditions/actions (UCE unsuccessfully)

2.1.1 Discovery

Sending entities that wish to use AMP SHOULD discover support for this protocol (using Service Discovery (XEP-0030) \(^3\)) along the intended path. Typically, this would involve sending disco#info queries to the sending entity’s own server and the server of the intended recipient. The results of these queries MAY be cached for up to 24 hours, unless otherwise expired.

If a server supports Advanced Message Processing, it MUST report that by including a service discovery feature of “http://jabber.org/protocol/amp” in the service discovery information result that it returns to the requesting entity.

Listing 1: Initial Service Discovery information request

```
<iq from='northumberland@shakespeare.lit/westminster'
     to='shakespeare.lit'
     type='get'>
   <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 2: Service Discovery information response

```
<iq from='shakespeare.lit'
     to='northumberland@shakespeare.lit/westminster'
```

A server SHOULD also maintain a service discovery node named "http://jabber.org/protocol/amp", at which it advertises the individual actions and conditions it supports. If an entity needs to determine whether the server supports individual actions and conditions, it SHOULD send a service discovery information request to that node; the server then MUST either return the list of supported actions and conditions or return an error such as <feature-not-implemented/>. (Note: If the server does not provide information for this disco node, the requesting entity MUST assume that all actions and conditions are supported for each reported action set or condition set.) Each supported action shall be reported as a feature using the following format:

http://jabber.org/protocol/amp?action={action}

Each supported condition shall be reported as a feature using the following format:

http://jabber.org/protocol/amp?condition={condition}

The following examples show the request-response flow for information about individual actions and conditions (note the inclusion of the 'node' attribute).

Listing 3: Request for information about individual actions and conditions

```xml
<iq from='northumberland@shakespeare.lit/westminster'
    to='shakespeare.lit'
    type='get'>
    <query xmlns='http://jabber.org/protocol/disco#info'
           node='http://jabber.org/protocol/amp'/>
</iq>
```

Listing 4: Response for individual actions and conditions

```xml
<iq from='shakespeare.lit'
    to='northumberland@shakespeare.lit/westminster'
    type='result'>
    <query xmlns='http://jabber.org/protocol/disco#info'
           node='http://jabber.org/protocol/amp'>
        <identity name='Shakespeare_IM-AMP_Support' category='im' type='server'/>
        ...
```
2.1.2 Specifying Semantics

Once support is determined, the sender can attach the desired delivery semantics to the message:

Listing 5: A message with AMP semantics

```xml
<message
    from='northumberland@shakespeare.lit'
    id='richard2-4.1.247'
    to='kingrichard@royalty.england.lit'>
    <body>My lord, dispatch; read o'er these articles.</body>
    <amp xmlns='http://jabber.org/protocol/amp'>
        <rule condition='expire-at' action='drop' value='2004-01-01T00:00:00Z'/>
    </amp>
</message>
```

The semantics are defined as a set of `<rule/>` elements within the `<amp/>` root element. Each `<rule/>` declares the condition to trigger on and the action to execute if triggered.

By default, the ruleset applies only to the "edge servers": those servers to which the sending and receiving entities are connected. (Note: For the purposes of Advanced Message Processing, "server" is defined as in XMPP Core and does not include any internal components, such as connection managers, that may provide functionality within a server implementation or installation.)

The ruleset MAY be applied to all server-to-server "hops" along the route from the sending and receiving entities by adding the "per-hop" attribute to the `<amp/>` element. The value of this attribute is either "true" (apply rules to all hops) or "false" (follow default behavior, i.e., apply rules at the edge servers only).

Listing 6: Another message with AMP semantics
For examples of validation failure, refer to the Error Handling section of this document. Note: Even if “per-hop” processing is requested, each server in the route MUST ignore rules that cannot apply to it; the Defined Conditions outline if they can be applied per-hop.

### 2.2 Server Processing

Server operation is where the bulk of the work is performed. Upon receiving a message with an AMP extension, the server performs the following flow:

1. Validate the semantics (E1, E2).
2. Determine the default behavior.
3. Process rules until condition is met.
   - If a condition is met, execute that rule’s action
   - If no conditions are met, perform “default” behavior for message
4. Execute determined action (UCE) (E3).
   - E1: The provided semantics (condition and/or action) are not supported or valid (UCE unsuccessfullly)
   - E2: The requested semantics are not acceptable (UCE unsuccessfullly)
   - E3: The next server does not support this protocol (UCE unsuccessfullly)

#### 2.2.1 Validating Semantics

Validation can take many forms, but at the very least the server MUST verify that it understands each of the rule conditions and actions, and that the condition contents are appropriate. The server MAY also refuse to accept certain combinations of conditions and actions, for example if they present a risk of violating security policies. If the semantics are
not valid, supported, or acceptable, the server MUST reply with an error specifying the rule(s) that are at issue. The server SHOULD validate all the semantics before returning an error. For syntax and examples of error handling related to validation failure, refer to the Error Handling section of this document.

2.2.2 Determine Default Action

This step is essentially what a server normally does, except that it does not actually perform the action. This determines what would happen to the message if there were no semantics attached (such as dispatch to another server or store offline). At this point, the server SHOULD also calculate any timing or calendar requirements (if applicable).

2.2.3 Process Rules

At this step, the server processes the attached semantics. The server MUST process the rules serially, and in the order they are presented within the <amp/> element. As soon as a rule’s condition is met, processing ends with that action overriding the default action determined earlier (unless the action permits continued processing).

2.2.4 Execute Determined Action

Once all rules have been processed or otherwise accounted for, the server executes the action determined at this point.

A server SHOULD NOT dispatch a <message/> stanza with AMP semantics to another server unless it knows that the next server supports AMP (this SHOULD be discovered via Service Discovery and MAY be cached to avoid delivery delays). If the next server does not support AMP, the current server replies to the original sender with a <service-unavailable/> error condition. Otherwise this flow starts again for the server to which the message has been dispatched.

2.2.5 Return Event

If the determined action involves returning an event (alert, error, or notification) to the sender, a server MUST send a <message/> stanza to the sender containing the rule that was met. This <message/> stanza MUST include the original value of the 'id' attribute and SHOULD NOT contain the non-AMP contents (e.g., <body/> child) originally included by the sender.
3 Conditions and Actions

The preceding sections of this document define the general behavior regarding AMP. This section outlines how `<rule/>` action and condition sets are specified. It also provides defined action and condition sets; these action and condition sets SHOULD be supported by any implementation of Advanced Message Processing, but support for any given action or condition set is not required. (Note: The action and condition sets defined herein may be supplemented in the future via registration of additional action and condition sets with the XMPP Registrar.)

3.1 Rule Condition Guidelines

The definition of a `<rule/>` condition MUST provide the following information:

- Define a namespace governing the condition set
- Define the condition:
  - Define the 'condition' attribute value
  - Specify if the "per-hop" flag applies
  - Define the syntax/format of the 'value' attribute value
  - Define how the "value" is interpreted to meet a condition

3.2 Rule Action Guidelines

The definition of a `<rule/>` action MUST provide the following information:

- Define a namespace governing the action set
- Define the action:
  - Define each 'action' attribute value
  - Define the expected behavior if the `<rule/>` is triggered

3.3 Defined Conditions

The condition defines how or when a particular rule is triggered. The value of the condition attribute determines what the contents of the `<rule/>` mean. The following conditions are defined by this document and SHOULD be supported by any implementation.

In the following sections, the terms "content" and "contents" refers to the XML character data contained by the `<rule/>` element.
3.3.1 deliver

The "deliver" condition is used to ensure delivery (or non-delivery) in one of five ways:

1. Directly to the specified address or account over XMPP
2. Delayed to offline storage (for later delivery via XMPP)
3. Forwarded to an alternate address or account over XMPP
4. Indirectly to an alternate address or account through a gateway to a non-XMPP system

This condition is met based on what the processor would do with the message at the moment of receipt, as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>direct</td>
<td>The message would be immediately delivered to the intended recipient or routed to the next hop.</td>
</tr>
<tr>
<td>forward</td>
<td>The message would be forwarded to another XMPP address or account.</td>
</tr>
<tr>
<td>gateway</td>
<td>The message would be sent through a gateway to an address or account on a non-XMPP system.</td>
</tr>
<tr>
<td>none</td>
<td>The message would not be delivered at all (e.g., because the intended recipient is offline and message storage is not enabled).</td>
</tr>
<tr>
<td>stored</td>
<td>The message would be stored offline for later delivery to the intended recipient.</td>
</tr>
</tbody>
</table>

This condition MAY be applied to each "hop" in the server route.

3.3.2 expire-at

The "expire-at" condition is used to ensure delivery before an absolute point in time. Naturally, this does not guarantee 4 that the message will not be delivered after that time from the sender's perspective, since this document does not assume that all machine clocks (e.g., for all servers along the delivery route) are synchronized. However, in order to help ensure that this condition is met correctly, servers that implement this document (or the machines that host such servers) SHOULD use the Network Time Protocol (RFC 1305 5) to keep in sync with established time authorities. Note also that expire-at functionality becomes less reliable the closer the expire-at time is to the present (e.g., the sender will receive less reliable delivery of a message specifying an expire-at time two seconds in the future than of a message specifying

---

4 Guarantee is a strong word. This document defines methods for making message delivery more reliable within certain bounds, but does not pretend that such methods provide any form of guaranteed delivery.

3 CONDITIONS AND ACTIONS

an expire-at time two hours or two days in the future).
The content of the ‘value’ attribute specifies some point after the exact moment the message is sent; the content MUST be a dateTime as specified in XMPP Date and Time Profiles (XEP-0082)\(^6\), and the timezone MUST be UTC.
The condition is met if the message would be delivered to the recipient after the specified datetime. To determine the datetime to compare to, the processor first determines if and when a message can be dispatched (e.g. not stored offline). The processor then records this datetime, and compares it with the specified datetime. If the current datetime is on or after that specified, the condition is met.
This condition MAY be applied to each ”hop” in the server route.

3.3.3 match-resource
The ”match-resource” condition is used to restrict delivery based on the resource identifier of the recipient JID.
The defined values for this condition are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>Destination resource matches any value, effectively ignoring the intended resource.</td>
<td>“home/laptop” matches “home”, “home/desktop” or “work/desktop”</td>
</tr>
<tr>
<td>exact</td>
<td>Destination resource exactly matches the intended resource.</td>
<td>“home/laptop” only matches “home/laptop” and not “home/desktop” or “work/desktop”</td>
</tr>
<tr>
<td>other</td>
<td>Destination resource matches any value except for the intended resource.</td>
<td>“home/laptop” matches “work/desktop”, “home” or “home/desktop”, but not “home/laptop”</td>
</tr>
</tbody>
</table>

The condition is met if the resource for the actual destination JID matches the intended JID using the above rules. For instance, if a message is intended for ”romeo@montague.net/work” with the ”match-resource” condition of ”exact”, the condition is met if the message can be immediately delivered only to ”romeo@montague.net/work”.
For purposes of this condition, an intended JID with no resource has the following behavior:

- If the value is ”exact”, the condition is met only if the server would deliver to a destination JID without a resource identifier (e.g., a Multi-User Chat (XEP-0045)\(^7\) room or offline storage).

---

3 CONDITIONS AND ACTIONS

• If the value is "other", the condition is met only if the server would not deliver to a destination JID without a resource identifier.

This condition MUST NOT be applied to each "hop" in the server route, only at the edge servers. If an <amp/> element includes this condition and also indicates that it should be processed per hop, this <rule/> shall be ignored.

Note: By design, this protocol does not include support for partial resource matching (which would stipulate, for example, that the resource identifiers "home/laptop" and "homeboy" both match "home").

3.4 Defined Actions

The action defines what occurs when a particular rule is triggered. The value of the action attribute determines the behavior if the rule’s condition is met.

The following actions are defined by this document.

3.4.1 alert

The "alert" action triggers a reply <message/> stanza to the sending entity. This <message/> stanza MUST contain the element <amp status='alert'/>, which itself contains the <rule/> that triggered this action. In all other respects, this action behaves as "drop".

Listing 7: Alert Response

```xml
<message from='hamlet.lit'
to='bernardo@hamlet.lit/elsinore'
id='chatty2'>
<amp xmlns='http://jabber.org/protocol/amp'
    status='alert'
    from='bernardo@hamlet.lit/elsinore'
    to='francisco@hamlet.lit'>
    <rule action='alert' condition='deliver' value='stored'/>
</amp>
</message>
```

3.4.2 drop

The "drop" action silently discards the message from any further delivery attempts and ensures that it is not placed into offline storage. The drop MUST NOT result in other responses.
3.4.3  error

The "error" action triggers a reply <message/> stanza of type "error" to the sending entity. The <message/> stanza’s <error/> child MUST contain a <failed-rules xmlns='http://jabber.org/protocol/amp#errors'/> error condition, which itself contains the rules that triggered this action.

Listing 8: Error Response

```xml
<message from='hamlet.lit'
to='bernardo@hamlet.lit/elsinore'
type='error'
id='chatty2'>
<amp xmlns='http://jabber.org/protocol/amp'
status='error'
to='bernardo@hamlet.lit/elsinore'
from='francisco@hamlet.lit'>
<rule action='error' condition='deliver' value='stored'/>
</amp>
<error type='modify' code='500'>
<undefined-condition xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
</error>
</message>
```

Note that the error SHOULD be of type "modify", and the general error condition SHOULD be <undefined-condition xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>.

3.4.4  notify

The "notify" action triggers a reply <message/> stanza to the sending entity. This <message/> stanza MUST contain the element <amp status='notify'/>, which itself contains the <rule/> that triggered this action. Unlike the other actions, this action does not override the default behavior for a server. Instead, the server then executes its default behavior after sending the notify.

Listing 9: Notify Response

```xml
<message from='hamlet.lit'
to='bernardo@hamlet.lit/elsinore'
id='chatty2'>
<amp xmlns='http://jabber.org/protocol/amp'
status='notify'
to='bernardo@hamlet.lit/elsinore'
/>
3.5 Description of Condition/Action Combinations

In general, a rule can be read as "do {action} if {condition} is true for {value}"; however, to facilitate understanding, this section describes the various condition/action combinations in plain English.

3.5.1 Deliver Rules

When the condition is "deliver", the rules can be described as follows:

<table>
<thead>
<tr>
<th>Action</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>direct</td>
<td>Return an alert if the message would be delivered directly.</td>
</tr>
<tr>
<td>alert</td>
<td>forward</td>
<td>Return an alert if the message would be forwarded to another XMPP address.</td>
</tr>
<tr>
<td>alert</td>
<td>gateway</td>
<td>Return an alert if the message would be delivered to a non-XMPP address through a gateway.</td>
</tr>
<tr>
<td>alert</td>
<td>none</td>
<td>Return an alert if no delivery is possible.</td>
</tr>
<tr>
<td>alert</td>
<td>stored</td>
<td>Return an alert if the message would be stored offline for later delivery via XMPP.</td>
</tr>
<tr>
<td>drop</td>
<td>direct</td>
<td>Drop the message if it would be delivered directly.</td>
</tr>
<tr>
<td>drop</td>
<td>forward</td>
<td>Drop the message if it would be forwarded to another XMPP address.</td>
</tr>
<tr>
<td>drop</td>
<td>gateway</td>
<td>Drop the message if it would be delivered to a non-XMPP address through a gateway.</td>
</tr>
<tr>
<td>drop</td>
<td>none</td>
<td>Drop the message if no delivery is possible.</td>
</tr>
<tr>
<td>drop</td>
<td>stored</td>
<td>Drop the message if it would be stored offline for later delivery via XMPP.</td>
</tr>
<tr>
<td>error</td>
<td>direct</td>
<td>Return an error if the message would be delivered directly.</td>
</tr>
<tr>
<td>error</td>
<td>forward</td>
<td>Return an error if the message would be forwarded to another XMPP address.</td>
</tr>
<tr>
<td>error</td>
<td>gateway</td>
<td>Return an error if the message would be delivered to a non-XMPP address through a gateway.</td>
</tr>
</tbody>
</table>
## Conditions and Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>none</td>
<td>Return an error if no delivery is possible.</td>
</tr>
<tr>
<td>error</td>
<td>stored</td>
<td>Return an error if the message would be stored offline for later delivery via XMPP.</td>
</tr>
<tr>
<td>notify</td>
<td>direct</td>
<td>Return a notification if the message would be delivered directly.</td>
</tr>
<tr>
<td>notify</td>
<td>forward</td>
<td>Return a notification if the message would be forwarded to another XMPP address.</td>
</tr>
<tr>
<td>notify</td>
<td>gateway</td>
<td>Return a notification if the message would be delivered to a non-XMPP address through a gateway.</td>
</tr>
<tr>
<td>notify</td>
<td>none</td>
<td>Return a notification if no delivery is possible.</td>
</tr>
<tr>
<td>notify</td>
<td>stored</td>
<td>Return a notification if the message would be stored offline for later delivery via XMPP.</td>
</tr>
</tbody>
</table>

### 3.5.2 Expire-at Rules

When the condition is "expire-at", the rules can be described as follows:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>Return an alert if the message would be delivered after the specified timestamp.</td>
</tr>
<tr>
<td>drop</td>
<td>Drop the message if it would be delivered after the specified timestamp.</td>
</tr>
<tr>
<td>error</td>
<td>Return an error if the message would be delivered after the specified timestamp.</td>
</tr>
<tr>
<td>notify</td>
<td>Return a notification if the message would be delivered after the specified timestamp.</td>
</tr>
</tbody>
</table>

### 3.5.3 Match-resource Rules

When the condition is "match-resource", the rules can be described as follows:

<table>
<thead>
<tr>
<th>Action</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>any</td>
<td>Return an alert if the message would be delivered to any resource.</td>
</tr>
<tr>
<td>alert</td>
<td>exact</td>
<td>Return an alert if the message would be delivered to the exact resource.</td>
</tr>
</tbody>
</table>
4 Formal Description

4.1 <amp/> Root Element

All delivery semantics are encapsulated in the <amp/> element. This element contains one or more <rule/> elements specifying the specific rules to process. It can optionally possess attributes about the current status, the original sender and recipient, and route applicability. The 'status' attribute specifies the reason for this <amp/> element. When specifying semantics to be applied (client to server), this attribute MUST NOT be present. When replying to a sending entity regarding a met condition, this attribute MUST be present and SHOULD be the value of the 'action' attribute for the triggered rule. (Note: Individual action definitions MAY provide their own requirements.)

The 'from' attribute specifies the original sender of the containing <message/> stanza. This attribute MUST be specified for any <message/> stanza sent from a supporting server, regardless of the recipient. It SHOULD NOT be specified otherwise. The value of the 'from' attribute MUST be the full JID (node@domain/resource) of the sender for the original <message/>
stanza.

The 'to' attribute specifies the original (intended) recipient of the containing <message/> stanza. This attribute MUST be specified for any <message/> stanza sent from a supporting server, regardless of the recipient. It SHOULD NOT be specified otherwise. The value of the 'to' attribute MUST be the full JID (node@domain/resource) of the intended recipient for the original <message/> stanza.

The 'per-hop' attribute flags the contained ruleset for processing at each server in the route between the original sender and original intended recipient. This attribute MAY be present, and MUST be either "true" or "false". If not present, the default is "false".

4.2 <rule/> Element

Each semantic rule is specified with a <rule/> element. This element possesses attributes for the condition, value, and action.

The 'action' attribute defines the result for this rule. This attribute MUST be present, and MUST be either a value defined in the Defined Actions section, or one registered with the XMPP Registrar.

The 'condition' attribute defines the overall condition this rule applies to. This attribute MUST be present, and MUST be either a value defined in the Defined Conditions section, or one registered with the XMPP Registrar.

The 'value' attribute defines how the condition is matched. This attribute MUST be present, and MUST NOT be an empty string (""). The interpretation of this attribute’s value is determined by the 'condition' attribute.

5 Example Scenarios

5.1 Reliable Data Transport

The <message/> stanza is nearly ideal for data transport, but to ensure reliability it is often desirable that such messages not be delivered to any resource but that specified. To facilitate this, the sending entity includes a <rule action='drop' condition='match-resource' value='other'/> (if failure notification is unnecessary) or <rule action='error' condition='match-resource' value='other'/> (if failure notification is required). The following example illustrates this using In-Band Bytestreams (XEP-0047) 8:

Listing 10: Sending a message for reliable data transport

```xml
<message to='francisco@hamlet.lit/pda'
         from='bernardo@hamlet.lit/elsinore'
         id='ibb1'>
  <data xmlns='http://jabber.org/protocol/ibb' sid='mySID' seq='0'>

```
5.2 Time-Sensitive Messages

Time-sensitive messages are a frequent occurrence in some environments (e.g., front-office personnel routinely notify others of "events" such as guests, unexpected refreshments, and ad-hoc gatherings). To send a time-sensitive message, the sending entity includes a `<rule action='drop' condition='expire-at'/>` with the time when the event is to occur:

Listing 12: Sending a time-sensitive message

```
<message to='linuxwolf@outer-planes.net'
         from='receptionist@outer-planes.net'>
```
5.3 Transient Messages

Transient messages are messages that should never be stored offline. To send a transient message, the sending entity includes a `<rule action='drop' condition='deliver' value='stored'/>`:

```
<message to='francisco@hamlet.lit'
    from='bernardo@hamlet.lit/elsinore'
    type='chat'
    id='chatty1'>
    <body>Who's there?</body>
    <amp xmlns='http://jabber.org/protocol/amp'>
        <rule action='drop' condition='deliver' value='stored'/>
    </amp>
</message>
```

Alternatively, the sending entity includes a `<rule action='alert' condition='deliver' value='stored'/>` to be alerted instead of having the message silently dropped:

```
<message to='francisco@hamlet.lit'
    from='bernardo@hamlet.lit/elsinore'
    type='chat'
    id='chatty2'>
    <body>Who's there?</body>
    <amp xmlns='http://jabber.org/protocol/amp'>
        <rule action='alert' condition='deliver' value='stored'/>
    </amp>
</message>
```
Listing 15: Sender alerted regarding transient message

```xml
<message from='hamlet.lit'
    to='bernardo@hamlet.lit/elsinore'
    id='chatty2'>
  <amp xmlns='http://jabber.org/protocol/amp'
    status='alert'
    from='bernardo@hamlet.lit/elsinore'
    to='francisco@hamlet.lit'>
    <rule action='alert' condition='deliver' value='stored'/>
  </amp>
</message>
```

6 Error Handling

6.1 Conditions

To simplify the discussion of error conditions, this document uses the following mappings between namespace URIs and namespace prefixes (note: this mapping is provided only for the purpose of simplifying the discussion and is not intended for use within the protocol itself).

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>xmpp</td>
<td>urn:ietf:params:xml:ns:xmpp-stanzas</td>
</tr>
<tr>
<td>msg</td>
<td><a href="http://jabber.org/protocol/amp">http://jabber.org/protocol/amp</a></td>
</tr>
<tr>
<td>err</td>
<td><a href="http://jabber.org/protocol/amp#errors">http://jabber.org/protocol/amp#errors</a></td>
</tr>
</tbody>
</table>

The following table shows the possible error conditions in relation to general AMP rules and conditions as well as the Defined Actions.
### General Condition Error Type Specific Condition Description

<table>
<thead>
<tr>
<th>General Condition</th>
<th>Error Type</th>
<th>Specific Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xmpp:bad-request/&gt;</code></td>
<td>modify</td>
<td><code>&lt;msg:unsupported-actions/&gt;</code></td>
<td>One or more rule's specified actions are not supported. The element <code>&lt;msg:unsupported-actions/&gt;</code> contains the <code>&lt;rule/&gt;</code> elements that specify the unsupported actions.</td>
</tr>
<tr>
<td><code>&lt;xmpp:bad-request/&gt;</code></td>
<td>modify</td>
<td><code>&lt;msg:unsupported-conditions/&gt;</code></td>
<td>One or more rule's specified conditions are not supported. The element <code>&lt;msg:unsupported-conditions/&gt;</code> contains the <code>&lt;rule/&gt;</code> elements that specify the unsupported conditions.</td>
</tr>
<tr>
<td><code>&lt;xmpp:not-acceptable/&gt;</code></td>
<td>modify</td>
<td><code>&lt;msg:invalid-rules/&gt;</code></td>
<td>One or more rules are not acceptable by this server, usually because the condition/action combination is restricted. The element <code>&lt;msg:invalid-rules/&gt;</code> contains the <code>&lt;rule/&gt;</code> elements that are not acceptable.</td>
</tr>
</tbody>
</table>
6.2 Examples

This section shows examples of the error conditions described in the previous section (for information regarding mapping of XMPP error conditions to Jabber error codes, refer to Error Condition Mappings (XEP-0086) ⁹).

6.2.1 Unsupported Action

Listing 16: A message with AMP semantics

```xml
<message>
  <from>'northumberland@shakespeare.lit'</from>
  <id>'richard2-4.1.247'</id>
  <to>'kingrichard@royalty.england.lit'></to>
  <body>My lord, dispatch; read o'er these articles.</body>
  <amp xmlns='http://jabber.org/protocol/amp'>
    <rule action='drop' condition='expire-at' value='2004-01-01T00:00:00Z'/>
  </amp>
</message>
```

6 ERROR HANDLING

Listing 17: Server does not support action

```xml
<message
  from='shakespeare.lit'
  id='richard2-4.1.247'
  to='northumberland@shakespeare.lit'
  type='error'>
  <amp xmlns='http://jabber.org/protocol/amp'>
    <rule action='drop' condition='expire-at' value='2004-01-01T00:00:00Z'/>
  </amp>
  <error type='modify' code='400'>
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
    <unsupported-actions xmlns='http://jabber.org/protocol/amp'>
      <rule condition='expire-at' action='drop' value='2004-01-01T00:00:00Z'/>
    </unsupported-actions>
  </error>
</message>
```

6.2.2 Unsupported Condition

Listing 18: A message with AMP semantics

```xml
<message
  from='shakespeare.lit'
  id='richard2-4.1.247'
  to='kingrichard@royalty.england.lit'>
  <body>My lord, dispatch; read o’er these articles.</body>
  <amp xmlns='http://jabber.org/protocol/amp'>
    <rule action='drop' condition='expire-at' value='2004-01-01T00:00:00Z'/>
  </amp>
</message>
```

Listing 19: Server does not support condition

```xml
<message
  from='shakespeare.lit'
  id='richard2-4.1.247'
  to='northumberland@shakespeare.lit'
  type='error'>
  <amp xmlns='http://jabber.org/protocol/amp'>
    <rule action='drop' condition='expire-at' value='2004-01-01T00:00:00Z'/>
  </amp>
  <error type='modify' code='400'>
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</message>
```
6.2.3 Not Acceptable

Listing 20: A message with AMP semantics

Listing 21: The rule is not acceptable to the server

6.2.4 Service Unavailable

Listing 22: A message with AMP semantics
Listing 23: AMP service is unavailable

Listing 24: A message with AMP semantics

Listing 25: Failed rules
8 Stream Feature

XMPP Core \(^{10}\) defines methods for advertising feature support during stream negotiation. For the sake of efficiency, it may be desirable for a server to advertise support for Advanced Message Processing as a stream feature. The namespace for reporting support within \(<\text{stream:features}/>\) is "http://jabber.org/features/amp". Upon receiving a stream header from the initiating entity, the receiving entity (usually a server) returns a stream header to initiating entity and MAY announce support for Advanced Message Processing registration by including the relevant stream feature:

Listing 26: Advertising Advanced Message Processing as a stream feature

```xml
<stream:features>
  <amp xmlns='http://jabber.org/features/amp'/>
</stream:features>
```

9 Security Considerations

Most AMP conditions could be used by unauthorized individuals to gain access to presence information about users of IM servers and other presence-based messaging systems. For example, consider the following scenario: the user <romeo@montague.net> is not an authorized subscriber to the presence of the user <nurse@capulet.com>, but sends a <message/> stanza with a "deliver" rule of "stored" and an action of "alert" to that address; if the Nurse is not online, Romeo would receive an AMP alert that the message has been stored offline, in the process discovering that the Nurse is offline. Similar scenarios are possible with the "match-resource" rule (send to the user’s usual resource) and the "expire-at" rule (send messages every minute to poll for presence). If a server implements presence subscriptions as specified in RFC 3921, it SHOULD NOT return alerts, errors, or other AMP notifications to senders who are not authorized to view the recipient’s presence information (via a subscription of "both" or "from") if the notification would reveal the recipient’s status as online or offline.

There are several directions server implementors can follow in this regard:

- Do not implement the relevant condition, as a result of which the server MUST reply with a <feature-not-implemented/> error condition if the condition is specified; however, this is unduly restricts the supported conditions and effectively cripples the AMP functionality, so is NOT RECOMMENDED.

- Accept the relevant condition only if the action is "drop", as a result of which the server MUST reply with a <not-acceptable/> error condition if the action is "alert", "error", or "notify"; this is slightly less restrictive but still unnecessarily restricts the functionality of the system, so is NOT RECOMMENDED.

- Accept the relevant condition only if the sender is authorized to receive the receiver’s presence, as a result of which the server MUST reply with a <not-acceptable/> error condition if the sender is not so authorized; this is the RECOMMENDED behavior.

10 IANA Considerations

No interaction with the Internet Assigned Numbers Authority (IANA) is necessary as a result of this document.

---

12 An exception might be fully-trusted or closed networks.
13 The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.
11 XMPP Registrar Considerations

11.1 Protocol Namespaces

The XMPP Registrar includes 'http://jabber.org/protocol/amp' and 'http://jabber.org/protocol/amp#errors' in its registry of protocol namespaces.

11.2 Stream Features

The XMPP Registrar includes the 'http://jabber.org/features/amp' namespace in its registry of stream feature namespaces.

11.3 Well-Known Service Discovery Nodes

The XMPP Registrar includes 'http://jabber.org/protocol/amp' in its registry of well-known Service Discovery nodes.

11.4 Registries

11.4.1 Rule Conditions Registry

The XMPP Registrar maintains a registry of AMP <rule/> conditions at <https://xmpp.org/registrar/amp-conditions.html>. In order to submit new values to this registry, the registrant shall define an XML fragment of the following form and either include it in the relevant XMPP Extension Protocol or send it to the email address <registrar@xmpp.org>:

```xml
<condition>
  <name>the value of the 'condition' attribute</name>
  <ns>the namespace to be used as a service discovery feature</ns>
  <per-hop>does the "per-hop" flag apply? [true|false]</per-hop>
  <value>
    The syntax (e.g., datatype or allowable values) of the 'value' attribute.
  </value>
  <processing>values that result in message processing</processing>
  <doc>the document (e.g., XEP) in which this condition is specified</doc>
</condition>
```

14 The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>. 

26
The registrant may register more than one condition at a time, each contained in a separate 
<condition/> element.
Note: The namespace for the condition set shall be included in the Service Discovery features registry.

```xml
<condition>
  <name>deliver</name>
  <ns:path>http://jabber.org/protocol/amp?condition=deliver</ns:path>
  <per-hop>true</per-hop>
  <value>[direct|forward|gateway|none|stored]</value>
  <processing>
    The condition is met if (1) the value is "direct" and the message can be immediately delivered or further dispatched, (2) the value is "forward" and the message can be forwarded to another XMPP address, (3) the value is "gateway" and the message can be sent to a non-XMPP address via a gateway, (4) the value is "none" and the message cannot be delivered at all, or (5) the value is "stored" and the message can be stored for later delivery.
  </processing>
  <doc>XEP-0079</doc>
</condition>

<condition>
  <name>expire-at</name>
  <ns:path>http://jabber.org/protocol/amp?condition=expire-at</ns:path>
  <per-hop>true</per-hop>
  <value>DateTime per XEP-0082</value>
  <processing>
    The condition is met if the message would be delivered after the specified DateTime.
  </processing>
  <doc>XEP-0079</doc>
</condition>

<condition>
  <name>match-resource</name>
  <per-hop>false</per-hop>
  <value>[any|exact|other]</value>
  <processing>
    The condition is met if (1) the value is "any" and the intended recipient has at least one available resource (as defined in the XMPP IM specification); (2) the value "exact" and the intended recipient has an available resource that exactly matches the JID specified in the 'to' address; (3) the value is "other" and the intended recipient has an available resource whose full JID is other than that specified in the 'to' address.
  </processing>
</condition>
```
11.4.2 Rule Actions Registry


In order to submit new values to this registry, the registrant shall define an XML fragment of the following form and either include it in the relevant XMPP Extension Protocol or send it to the email address <registrar@xmpp.org>:

```xml
<action>
  <name>the value of the 'action' attribute</name>
  <ns>the namespace to be used as a service discovery feature</ns>
  <behavior>the expected behavior if the rule is triggered</behavior>
  <doc>the document (e.g., XEP) in which this action is specified</doc>
</action>
```

The registrant may register more than one action at a time, each contained in a separate <action/> element.

Note: The namespace for the action set shall be included in the Service Discovery features registry.

```xml
<action>
  <name>alert</name>
  <ns>http://jabber.org/protocol/amp?action=drop</ns>
  <behavior>
    The message is silently discarded but an alert is returned to the sender.
  </behavior>
  <doc>XEP-0079</doc>
</action>

<action>
  <name>drop</name>
  <ns>http://jabber.org/protocol/amp?action=drop</ns>
  <behavior>
    The message is silently discarded.
  </behavior>
  <doc>XEP-0079</doc>
</action>

<action>
  <name>error</name>
</action>
```
<ns\path{http://jabber.org/protocol/amp?action=error}</ns>

<behavior>
The message is not processed and an error is returned to the sender, specifying which rule resulted in failed processing.
</behavior>
<doc>XEP-0079</doc>
</action>

<action>
  <name>notify</name>
<ns\path{http://jabber.org/protocol/amp?action=notify}</ns>
<behavior>
The message is processed and a notification message is returned to the sender, specifying which rule was processed.
</behavior>
<doc>XEP-0079</doc>
</action>

12 XML Schemas

12.1 AMP

<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='http://jabber.org/protocol/amp'
  xmlns='http://jabber.org/protocol/amp'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in XEP-0079: http://www.xmpp.org/extensions/xep-0079.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='amp'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='rule' minOccurs='1' maxOccurs='unbounded'/>
      </xs:sequence>
      <xs:attribute name='from' use='optional' type='xs:string'/>
      <xs:attribute name='per-hop' use='optional' type='xs:boolean'
          default='false'/>
      <xs:attribute name='status' use='optional' type='xs:NCName'/>
    </xs:complexType>
  </xs:element>

</xs:schema>
12.2 Errors

```xml
<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='http://jabber.org/protocol/amp#errors'
    xmlns='http://jabber.org/protocol/amp#errors'
    elementFormDefault='qualified'>

    <xs:element
        name='rule'
        type='xs:NCName'/>

    <xs:element
        name='invalid-rules'
        type='xs:NCName'/>

    <xs:element
        name='unsupported-actions'
        type='xs:NCName'/>

    <xs:element
        name='unsupported-conditions'
        type='xs:NCName'/>

</xs:schema>
```
12.3 Stream Feature

```xml
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
 xmlns:xs='http://www.w3.org/2001/XMLSchema'
 targetNamespace='http://jabber.org/features/amp'
 xmlns='http://jabber.org/features/amp'
 elementFormDefault='qualified'>
  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in XEP-0079: http://www.xmpp.org/extensions/xep-0079.html
    </xs:documentation>
  </xs:annotation>
  <xs:element name='amp' type='empty'/>
  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value=''/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```
13 Acknowledgements

Thanks to Marshall Rose and Craig Kaes for their comments.