XEP-0001: XMPP Extension Protocols

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

Dave Cridland
mailto:dave@hellopando.com
xmpp:dwd@dave.cridland.net

Ralph Meijer
mailto:ralphm@ik.nu
xmpp:ralphm@ik.nu

2024-03-11
Version 1.25.0

Status	Type	Short Name
Active	Procedural	N/A

This document defines the standards process followed by the XMPP Standards Foundation.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

The XMPP Standards Foundation (XSF) [1] adheres to an open standards process that enables interested parties to document existing protocols used within the Jabber/XMPP developer community and to submit proposals that define new protocols; with a few exceptions, [2] such protocols can be considered extensions to the Extensible Messaging and Presence Protocol (XMPP) approved by the Internet Engineering Task Force (IETF) [4] in XMPP Core [5] and XMPP IM [6]. The focal point of the process is a series of protocol specifications called XMPP Extension Protocols or XEPs. [7]
Advancement of a XEP through the XSF's standards process is contingent on three factors:

- Rough consensus on the XSF's public discussion lists.
- Running code in XMPP clients, servers, and libraries.
- Formal approval by the XMPP Council [11].

The XSF's standards process can be outlined informally as follows:

1. A developer in the XMPP community defines a new XMPP extension that solves an existing problem or enables an innovative feature that is not addressed in the current XMPP protocol stack.
2. The developer submits a specification to the XMPP Extensions Editor [12] and agrees to transfer ownership over the protocol (but not implementations thereof) to the XSF.
3. If the specification is accepted by the XMPP Council, it is published as an Experimental XEP.

[1] The XMPP Standards Foundation (XSF) is an independent, non-profit membership organization that develops open extensions to the IETF's Extensible Messaging and Presence Protocol (XMPP). For further information, see <https://xmpp.org/about/xmpp-standards-foundation>.
[2] Effectively the only such exceptions are protocols that were superseded by RFC 3920 [3] and RFC 3921.
[4] The Internet Engineering Task Force is the principal body engaged in the development of new Internet standard specifications, best known for its work on standards such as HTTP and SMTP. For further information, see <http://www.ietf.org/>.
[5] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[6] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.
[7] The JEP (now XEP) concept as exemplified in version 1.0 of this document (approved in July of 2001) was borrowed from the Python community (see PEP-1). Subsequent revisions have been based on the Jabber/XMPP developer community's experience with this standards process, as well as insights gleaned from the standards processes followed by the IETF (RFC 2026 [8]), the World Wide Web Consortium (W3C) [9] (W3C Process Document [10]), and other standards development organizations. (Note: The term "XEP" is normally pronounced "zepp".)
[11] The XMPP Council is a technical steering committee, authorized by the XSF Board of Directors and elected by XSF members, that approves of new XMPP Extensions Protocols and oversees the XSF's standards process. For further information, see <https://xmpp.org/about/xmpp-standards-foundation#council>.
[12] The XMPP Extensions Editor is the individual appointed by the XSF Board of Directors to handle protocol submissions and provide day-to-day management of the XSF's standards process. For further information, see <https://xmpp.org/about/xsf/editor-team>.

4. The XEP undergoes extensive community review on the XSF's open discussion lists and is implemented experimentally in XMPP clients, servers, and libraries.
5. If the protocol fills an important need, meets with rough consensus, and is (preferably) implemented in running code, the XMPP Council will formally review it and vote on advancing it to a status of Stable.
6. While the XEP is in the Stable state, the protocol may be refined based on further discussion and implementation experience.
7. If the protocol is deemed stable after at least six months (and normally more) of further implementation and deployment experience, the XSF then considers advancement of the XEP to a status of Final.

Other types of XEP follow modified variants through the same essential path.
The remainder of this document formally defines the nature of XMPP Extension Protocols and the mechanisms for managing and advancing them within the XSF's standards process.

2 Objectives

The XMPP Standards Foundation was founded in the year 2001 to openly document, safeguard, manage, and extend the wire protocols used within the XMPP developer community. The work of the XMPP Standards Foundation has several objectives:

1. To produce practical, technically excellent solutions to important problems of real-time communication based on the set of streaming XML technologies known as XMPP.
2. To document XMPP extensions in a clear, concise manner so that the task of implementing the protocols is straightforward.
3. To ensure interoperability among the disparate technologies used on XMPP networks.
4. To guarantee that any person or entity can implement the protocols without encumbrance.
5. To work in an fair, open, objective manner.

The standards process specified herein has been developed and refined in order to meet these objectives.

3 XEP Types

The five XEP types are described in the following sections.
The approving body for all Standards Track, Informational, and Historical XEPs is the XMPP Council; the approving body for Humorous XEPs is the XMPP Extensions Editor; and the approving body for Procedural XEPs may be either the XSF Board of Directors [13] or the XMPP Council.
This document focuses primarily on Standards Track XEPs since they are the vehicle for defining new protocols, but also discusses the other XEP types.

3.1 Standards Track

A Standards Track XEP defines one of the following:

1. A wire protocol intended to be used as a standard part of XMPP technologies. [14]
2. A protocol suite that determines conformance requirements (e.g., XMPP Compliance Suites 2010 (XEP-0270) [15]).

This document concentrates on this type of XEP; variances from the procedures for Standards Track XEPs are noted where appropriate.

3.2 Informational

An Informational XEP typically defines best practices for implementation or deployment of an existing protocol (e.g., Service Discovery Extensions (XEP-0128) [16] and Best Practices for Use of SASL ANONYMOUS (XEP-0175) [17]).

3.3 Historical

An Historical XEP documents a protocol that was developed before the XSF's standards process was instituted, but that is still in use within the XMPP developer community; such a XEP may or may not be obsoleted by a Standards Track XEP, or upgraded to Standards Track.

[13] The XSF Board of Directors is an elected body that possesses overall responsibility for the affairs of the XMPP Standards Foundation. For further information, see <https://xmpp.org/about/xmpp-standards-foundation#board>.
[14] A protocol defined in a Standards Track XEP is not considered a full standard of the XMPP Standards Foundation until it achieves a status of Final within the standards process defined herein (a Standards Track XEP that has achieved a status of Stable may be referred to as a Stable Standard; a Standards Track XEP that has a status of Experimental must not be referred to as a standard, but instead should be referred to as a work in progress).
[15] XEP-0270: XMPP Compliance Suites 2010 <https://xmpp.org/extensions/xep-0270.html>.
[16] XEP-0128: Service Discovery Extensions <https://xmpp.org/extensions/xep-0128.html>.
[17] XEP-0175: Best Practices for Use of SASL ANONYMOUS <https://xmpp.org/extensions/xep-0175.html>.

3.4 Humorous

A Humorous XEP attempts to be funny by defining a protocol that would never be used in the real world; such XEPs are usually published on April 1 and automatically have a status of Active.
Note that the Approving Body for Humorous XEPs is the XMPP Extensions Editor itself.

3.5 Procedural

A Procedural XEP defines a process or activity to be followed by the XSF (e.g., XMPP Registrar Function (XEP-0053) [18]), including SIG charters as specified by Special Interest Groups (XEP-0002) [19].
For some Procedural XEPs, the Approving Body is the XSF Board instead of the XMPP Council.

4 Submission Process

The XSF welcomes and encourages the submission of protocols to the XSF's standards process. [20] Any individual or group of individuals may author a proposal and submit it to the XSF for consideration as a XEP, and there is no requirement that a XEP author shall be an elected member of the XSF. Proposals to define official XSF protocols must be presented in the XEP format and must follow the rules defined herein (after a proposal has been submitted but before it has been accepted as a XEP, it is known informally as a "ProtoXEP"). The authoring and submission process is defined in Guidelines for Authors of XMPP Extension Protocols (XEP-0143) [21]. All submissions to the XSF's standards process should be directed to the XMPP Extensions Editor.
Note well that XEP authors must transfer ownership of their protocols (but not implementations thereof) to the XSF. Refer to the XSF IPR Policy [22] for details. XEP authors must make sure that they have read, understood, and agreed to the XSF IPR Policy before submitting a proposal to the XMPP Extensions Editor!
All proposals submitted to the XSF for consideration as XEPs must contain the following information:

[18] XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.
[19] XEP-0002: Special Interest Groups <https://xmpp.org/extensions/xep-0002.html>.
[20] It is important to understand that private extensions to XMPP are also allowed. The XSF does not, and cannot, require such private extensions to be added to the public, official set of protocols recognized by the XSF. The processes and procedures in this document apply only to protocols that are submitted to the XSF, not to private protocol extensions used for custom functionality in particular applications. However, such private extensions must not be considered part of the protocols recognized by the XSF.
[21] XEP-0143: Guidelines for Authors of XMPP Extension Protocols <https://xmpp.org/extensions/xep-0143.html>.
[22] The XSF IPR Policy defines the XMPP Standards Foundation's official policy regarding intellectual property rights (IPR) as they pertain to XMPP Extension Protocols (XEPs). For further information, see <https://xmpp.org/about/xsf/ipr-policy>.

1. Legal Notices -- the legal notices must be exactly those specified in the XSF IPR Policy

2. Author Information -- first name, last name, email address, and Jabber ID are all required and must be provided for all authors (to simplify the text we use singular "author" in the remainder of this document, with the understanding that there can be multiple authors)

Finally, Standards Track, Informational, and Historical XEPs must conform to RFC 2119 [23] in the use of terminology regarding requirements levels.

5 Publication Process

The Approving Body for almost all XEPs is the XMPP Council; therefore, in order to be published as a XEP, a proposal must first be accepted by the XMPP Council (the only exceptions are certain kinds of Procedural XEPs, for which the approving body may be the XSF Board of Directors and which may be accepted for publication by the XMPP Extensions Editor in consultation with the Board). Upon receiving a proposal, the XMPP Extensions Editor shall do the following:

- ensure that its format is correct
- publish it to <http://www.xmpp.org/extensions/inbox/>
- publicly announce its existence by sending a message to the discussion list of the Standards SIG [24]
- request acceptance of the proposal as a XEP by the Approving Body (normally the XMPP Council)

Within 14 days, the Chair of the Approving Body shall poll members of the Approving Body. In rare cases, the Approving Body may decide that the wrong Approving Body may have been chosen. If the Approving Body is in doubt, the XSF Board shall determine the correct Approving Body.
If no member of the Approving Body objects to publication of the proposal, the XMPP Extensions Editor shall accept it as a XEP. The precise rules for polls, including meetings and timeouts, are determined by the Chair of the Approving Body.
If objections are raised by the Approving Body on the discussion list of the Standards SIG [25]

[23] RFC 2119: Key words for use in RFCs to Indicate Requirement Levels <http://tools.ietf.org/html/rfc2119>.
[24] The Standards SIG is a standing Special Interest Group devoted to development of XMPP Extension Protocols. The discussion list of the Standards SIG is the primary venue for discussion of XMPP protocol extensions, as well as for announcements by the XMPP Extensions Editor and XMPP Registrar. To subscribe to the list or view the list archives, visit <https://mail.jabber.org/mailman/listinfo/standards/>.
[25] The Standards SIG is a standing Special Interest Group devoted to development of XMPP Extension Protocols. The discussion list of the Standards SIG is the primary venue for discussion of XMPP protocol extensions, as well as for announcements by the XMPP Extensions Editor and XMPP Registrar. To subscribe to the list or view the list archives, visit <https://mail.jabber.org/mailman/listinfo/standards/>.

(the Standards list) or in its meeting, the XEP author is encouraged to address the feedback of the Approving Body and to submit a revised version of the proposal and/or confer with the XMPP Extensions Editor or objecting Approving Body member(s) regarding how to proceed.
If the proposal is accepted as a XEP, the XMPP Extensions Editor shall do the following:

- assign it a number
- specify an appropriate type
- specify a status of Experimental (or Active for Humorous XEPs)
- add it to source control [26]
- add tracking information to the XEPs database
- publish version 0.1 of the XEP to the xmpp.org website [27]
- publicly announce the existence of the XEP by sending a message to the Standards list

Note well that no special criteria (other than acceptance by the Approving Body and minimal formatting compliance) need to be met in order for a XEP to be granted a status of Experimental. The granting of Experimental status must not be construed as indicating any level of approval by the XSF, the XMPP Council, or the XMPP developer community. Implementation of Experimental XEPs is encouraged in an exploratory fashion (e.g., in a proof of concept) in order to gain experience with and iteratively improve the protocol defined therein, but such implementations might not be appropriate for deployment in production systems.

6 Discussion Process

Once a XEP is published, it becomes available for public discussion within the Standards SIG and the broader XMPP developer community. The XEP author (or Document Shepherd) is responsible for collecting feedback from the XMPP developer community during the life of the XEP and for incorporating such feedback into the proposal. In order to fully participate in discussion of the proposal, they should be subscribed to the Standards list, which is the primary venue for discussion of XMPP Extension Protocols. Changes made based on feedback received by the XEP author (or Document Shepherd) shall be captured in updated versions of the XEP (e.g., 0.2 after 0.1), each of which must be put under source control and subsequently published and announced by the XMPP Extensions Editor.
If an Experimental XEP is inactive (i.e., no updated versions are published) for a period of twelve (12) months, the XMPP Extensions Editor shall automatically change the status of the

[26] XEPs are kept under source control in the 'xmpp' module and 'extensions' directory of the XSF Git repository; instructions for accessing these files can be found at <http://xmpp.org/about-xmpp/xsf/xsf-source-control/>.
[27] The canonical URL for accessing XMPP Extensions is <http://www.xmpp.org/extensions/>.

XEP to Deferred unless it is in the queue of XEPs under active consideration for advancement by the Approving Body; upon submission of an updated version, the XMPP Extensions Editor shall change the status back to Experimental.
Only substantial, non-editorial changes (e.g. those that would cause an updated version of 0.1 to 0.2, not editorial updates from 0.1.1 to 0.1.2) count as *activity* (or *updates*) for the purpose of considering moving a XEP from or to Deferred state.

7 Proposal Process

An Experimental (or Deferred) XEP may be proposed to the Approving Body for advancement to Stable (Standards Track XEPs) or Active (Historical, Informational, and Procedural XEPs). This can be requested from the Approving Body on the Standards list by, or in collaboration with, the XEP author. In case the XEP has been abandoned by its author(s), any other individual can propose advancement in their stead. The Approving Body will then require a Document Shepherd to take on responsibilities on behalf of the XEP author during the proposal and approval processes. The Approving Body must agree that the XEP is ready to be considered for advancement. Once the Approving Body so agrees, it shall instruct the XMPP Extensions Editor to (1) change the status of the XEP from Experimental (or Deferred) to Proposed and (2) issue a Last Call for open discussion on the Standards list. The Last Call shall expire not less than fourteen (14) days after the date of issue.
Once the consensus of the Standards SIG has been incorporated into the XEP and all issues of substance raised during the Last Call have been addressed by the XEP author or Document Shepherd, the XMPP Extensions Editor shall formally propose a specific revision of the XEP to the Approving Body for its vote. If necessary, the XMPP Extensions Editor may, at his discretion and in consultation with the Approving Body, extend the Last Call or issue a new Last Call if the XEP requires further discussion.

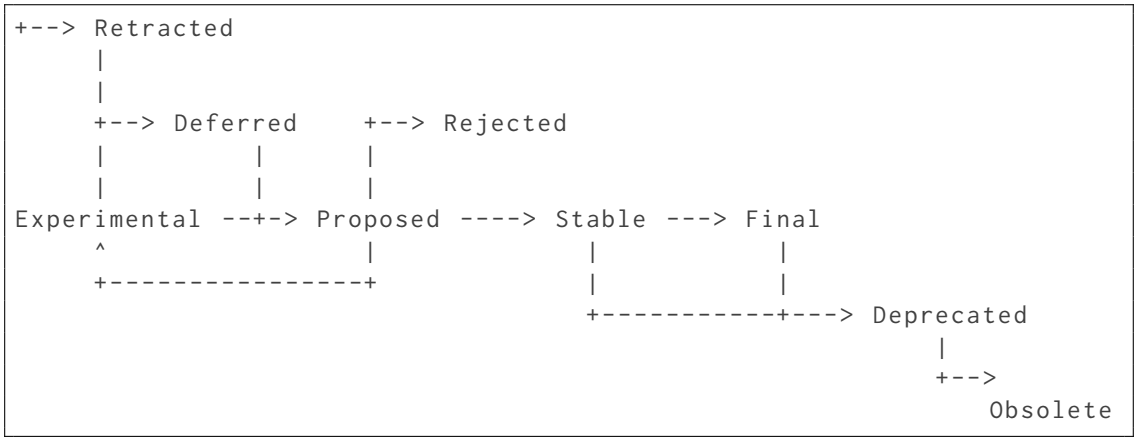
8 Approval Process

The precise mechanism for approval depends on the Approving Body.
After a XEP has been proposed to the XMPP Council, any change in its status shall be determined by a vote of the XMPP Council. All members of the Council must vote, with the possible values being +1 (approve), 0 (neutral), or -1 (disapprove, with reasons). A XEP shall not be advanced to the next stage in the approval process so long as any Council Member continues to vote -1; that Council Member's written concerns must be addressed in order for the XEP to advance. A majority of Council members must vote +1 in order for a XEP to advance. (Additional voting policies, such as voting periods and defaults if a member does not vote, may be set by the XMPP Council.) A vote of the XMPP Council is final and binding, although a XEP author or Document Shepherd is free to address the concerns of the Council and to resubmit the XEP for future consideration.
If the Approving Body decides after Last Call that the XEP is not ready for advancement yet,

but nevertheless useful, it can vote to move it back to Experimental state.
If the XMPP Council does not complete voting on a XEP before the end of its term, the XMPP Extensions Editor shall issue a new Last Call on the Standards list and the newly-elected Council shall vote anew on the XEP after completion of the Last Call. This provides an opportunity for any member of the previous Council who had voted -1 to voice his or her concerns in a public forum before the new Council votes on the XEP.
A vote of the XMPP Council applies only to the specific revision of the XEP that has been presented to it. Further revisions may need to be re-submitted for approval.
Any change in the status of a XEP must be announced on the Standards list by the XMPP Extensions Editor. If a XEP advances to a status of Final, it shall be so announced and also published as one of the official XSF protocols of the XMPP Standards Foundation.
Approval of Procedural XEPs for which the approving body is the XSF Board of Directors shall occur upon approval by the Board in accordance with the rules defined in the XSF Bylaws [28].
Approval for Humorous XEPs (for which the Approving Body is the XMPP Extensions Editor) is automatic upon accepting the submission.
More detailed information about the approval process is provided below, including criteria for Standards Track XEPs and for Historical, Informational, and Procedural XEPs.

8.1 Standards Track XEPs

The possible states for a Standards Track XEP are as follows:



After an XMPP Extension Protocol has been accepted for publication by the XMPP Council and before it is proposed for advancement to a status of Stable (or retracted or deferred), it shall have a status of Experimental. Publication as an Experimental XEP does not indicate approval of the protocol by the XMPP Council or the broader XMPP community.
Note: An Experimental specification is a work in progress and may undergo significant modification before advancing to a status of Stable. While implementation of an Experimental protocol is encouraged

[28] The Bylaws of the XMPP Standards Foundation (XSF) define the legal basis and operating procedures of the XSF. For further information, see <https://xmpp.org/about/xsf/bylaws>.

in order to determine the feasibility of the proposed solution, it is not recommended for such implementations to be included in the primary release for a software product (as opposed to an experimental branch).

The ideal path is for a Standards Track XEP is to be advanced by the XMPP Council from Proposed to Stable to Final (the criteria for this advancement are described in the following paragraphs). However, an Experimental XEP shall be assigned a status of Deferred if it has not been updated in twelve (12) months (e.g., because of a lack of interest or because it depends on other specifications that have yet to move forward). In addition, rather than being advanced from Proposed to Stable, a Standards Track XEP may be voted to a status of Rejected if the XMPP Council deems it unacceptable. (Note that if a XEP is Deferred, the XMPP Extensions Editor may at some point re-assign it to Experimental status, and that, even if a XEP is Rejected, it is retained in source control and on the XMPP Standards Foundation website for future reference.) Finally, (only) a XEP author may voluntarily remove an Experimental XEP from further consideration, resulting in a status of Retracted.

In order for a Standards Track XEP to advance from Proposed to Stable, it must:

1. fill known gaps in XMPP technologies or deficiencies with existing protocols
2. be clearly described and accurately documented so that it can be understood and implemented by interested and knowledgeable members of the XMPP developer community
3. document any known security considerations with the proposed technology
4. be generally stable and appropriate for further field experience
5. have achieved rough consensus (though not necessarily unanimity) within the Standards SIG
6. be formally defined by an XML schema
7. receive the requisite votes from the XMPP Council

Elevation to Stable status (version 1.0) is a major advancement for the XEP, indicating a strong sense on the part of the XMPP Council and XMPP developer community that the specification will be of lasting value.

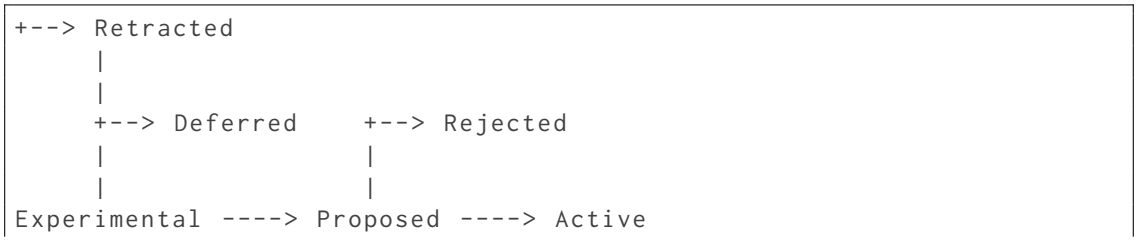
Note: Once an XMPP Extension Protocol has been advanced to a status of Stable, it is expected that the specification will be the basis for widespread implementation and for deployment in production environments. As a result of such implementation and deployment experience, the protocol may be subject to modification, including changes that are backwards-incompatible. Although such backwards-incompatible modifications shall be avoided if at all possible, deployment of a Stable protocol in mission-critical application may not be advisable.

Any changes to a Stable XEP that could reasonably be construed as material must be provisionally published, announced and discussed on the Standards mailing list, and formally approved by the XMPP Council before being officially published at the canonical URL for the XEP. Ultimate authority for Stable XEPs rests with the XMPP Council, which can at its discretion demand the reversal of any changes made by the XMPP Extensions Editor or the

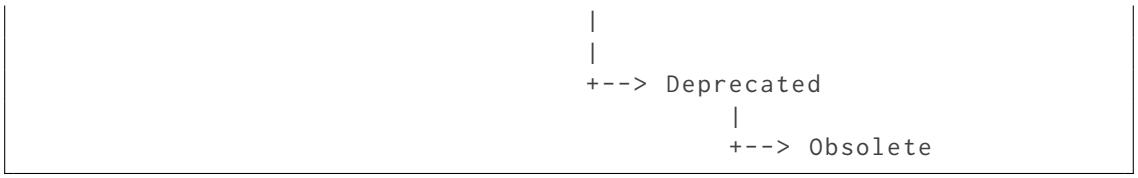
XEP author while the XEP is in the Stable state.
In order for a XEP to advance from Stable status to Final status (version 2.0), it must be shown to be stable and well-received by the XMPP developer community. Before presenting a Stable standard to the XMPP Council for consideration as a Final standard, the XMPP Extensions Editor shall issue a Call for Experience on the Standards list so that feedback can be gathered from those who have implemented the Stable standard (the Call for Experience shall expire not less than fourteen (14) days after the date of issue, and shall not be issued until at least six (6) months have passed since advancement to Stable). In addition, at least two implementations of the XEP must exist, at least one of which must be free software (in accordance with the The General Public License [29] or The Lesser General Public License [30]) or open-source software (in accordance with the definition provided by The Open Source Initiative [31]). Until two implementations are produced, a Standards Track XEP shall retain a status of Stable. Once (1) two implementations have been presented to the XMPP Council, (2) feedback provided during the Call for Experience has been incorporated into the XEP, and (3) the XEP has been fully checked for accuracy, the status of the XEP may be changed to Final upon a vote of the Council.
Note: Once an XMPP Extension Protocol has been advanced to a status of Final, every effort shall be made to limit the scope of modifications; in particular, backwards-incompatible changes shall not be made. However, limited modifications may be made as long as they are optional, backwards-compatible extensions rather than modifications to the core protocol itself. Therefore, a Final protocol is safe for deployment in mission-critical applications.
A Standards Track XEP that has been advanced to a status of Final may be superseded by a future XEP approved by the XMPP Council. In such cases, the status of the earlier XEP shall be changed to Deprecated, possibly with an expiration date assigned by the XMPP Council (see the Expiration Dates section below). After a reasonable period of time or upon the passing of the expiration date, the status of the XEP shall be changed to Obsolete.

8.2 Historical, Informational, and Procedural XEPs

The possible states for a Historical, Informational, or Procedural XEP are as follows:



[29] The General Public License is the primary code license for free software as defined by the Free Software Foundation. For further information, see <http://www.gnu.org/licenses/gpl.txt>.
[30] The Lesser General Public License is a secondary code license for free software as defined by the Free Software Foundation. For further information, see <http://www.gnu.org/licenses/lgpl.txt>.
[31] The Open Source Initiative defines the term 'open source' and maintains a list of open-source code licenses. For further information, see <http://www.opensource.org/>.



Because such XEPs do not seek to define standard protocols, in general they are less controversial and tend to proceed from Proposed to Active without controversy on a vote of the XMPP Council. However, some of these XEPs may be remanded from the Council to the XEP author and/or XMPP Extensions Editor for revision in order to be suitable for advancement from Proposed to Active (e.g., documentation of protocols in use must be accurate and describe any existing security concerns). As with Standards Track XEPs, the XEP author may retract such a XEP when it is Experimental, and the Council may reject such a XEP when it is Proposed.
Once approved, Historical, Informational, and Procedural XEPs will have a status of Active. Such a XEP may be replaced by a new XEP on the same or a similar topic, thus rendering the earlier XEP out of date; in such cases, the earlier XEP shall be assigned a status of Deprecated (and eventually Obsolete) with a note specifying the superseding XEP.
The XMPP Council may, at its discretion, decide to convert an Historical XEP into a Standards Track XEP if the protocol defined in the XEP has been in long use, is deemed stable and uncontroversial, and is unlikely to be superseded by a newer protocol. The Historical XEP shall be treated in the same way as a Standards Track XEP that has a status of Experimental, beginning with the Proposal Process. If after the Last Call and voting by the XMPP Council the XEP is approved for advancement on the standards track, its type shall be changed to Standards Track and its status shall be changed to Stable.

9 Summary of XEP States

The possible states for a XEP are summarized in the following sections.

9.1 Experimental

A XEP of any type is in the Experimental state after it has been accepted by the XMPP Council and published by the XMPP Standards Foundation but before it has advanced within the standards process to a state of Active or Stable.
Note: An Experimental specification is a work in progress and may undergo significant modification before advancing to a status of Stable. While implementation of an Experimental protocol is encouraged in order to determine the feasibility of the proposed solution, it is not recommended for such implementations to be included in the primary release for a software product (as opposed to an experimental branch).

9.2 Proposed

A XEP of any type is in the Proposed state while it is in Last Call or under consideration by the XMPP Council for advancement from Experimental to Stable or Active.

9.3 Draft

Note: In previous versions of the XSF process the "Stable" status was called "Draft". This led to wide spread confusion about the stability of the protocol so the status was renamed to better reflect the intent.

9.4 Stable

A Standards Track XEP is in the Stable state after it has undergone extensive discussion and technical review on the Standards list and has been voted forward on the standards track by the XMPP Council.
Note: Once an XMPP Extension Protocol has been advanced to a status of Stable, it is expected that the specification will be basis for widespread implementation and for deployment in production environments. As a result of such implementation and deployment experience, the protocol may be subject to modification, including changes that are backwards-incompatible. Although such backwards-incompatible modifications shall be avoided if at all possible, deployment of a Stable protocol in mission-critical application may not be advisable.

9.5 Final

A Standards Track XEP is in the Final state after it has been in the Stable state for at least six (6) months, has been implemented in at least two separate codebases, and has been voted forward on the standards track by the XMPP Council.
Note: Once an XMPP Extension Protocol has been advanced to a status of Final, every effort shall be made to limit the scope of modifications; in particular, backwards-incompatible changes shall not be made. However, limited modifications may be made as long as they are optional, backwards-compatible extensions rather than modifications to the core protocol itself. Therefore, a Final protocol is safe for deployment in mission-critical applications.

9.6 Active

A XEP of any type other than Standards Track is advanced to a status of Active after it has been voted forward from Experimental by the XMPP Council.

9.7 Deferred

An Experimental XEP of any type is changed to the Deferred state if it has not been updated in twelve (12) months.

9.8 Retracted

A XEP of any type is in the Retracted state if the author has asked the XMPP Extensions Editor to remove the XEP from further consideration in the XSF's standards process.

9.9 Rejected

A XEP of any type is in the Rejected state if the XMPP Council has deemed it unacceptable and has voted to not move it forward within the standards process.

9.10 Deprecated

A XEP of any type is in the Deprecated state if the XMPP Council has determined that the protocol defined therein is out of date and that new implementations are no longer encouraged (e.g., because it has been superseded by a more modern protocol).

9.11 Obsolete

A XEP of any type is changed from Deprecated to Obsolete if the XMPP Council has determined that the protocol defined therein should no longer be implemented or deployed.

10 Modification of Final and Active XEPs

Sometimes it is necessary to modify XEPs that have received final approval by the XMPP Council or XSF Board of Directors (e.g., to correct errors, incorporate the lessons of experience, or document new security concerns). This section describes the process for doing so with regard to Standards Track XEPs that have achieved a status of Final and Historical, Informational, and Procedural XEPs that have achieved a status of Active.
With regard to Standards Track XEPs, the XMPP Standards Foundation (in particular, the XMPP Council) strives to ensure that such XEPs are accurate, complete, and stable before advancing them to a status of Final (corresponding to document version 2.0 of the XEP). The Call for Experience and discussion within the Standards SIG help to ensure this result, but final responsibility rests with the XMPP Council. Despite the best efforts of all concerned, errors are sometimes discovered in Final XEPs (the individual who discovers such an error should inform the Council via the Standards mailing list or communicate directly with the

XMPP Extensions Editor). Whereas other standards development organizations may issue errata while leaving the specification itself unchanged, the XSF makes changes to the Final XEP "in place", where any changes that could reasonably be construed as material are subject to review by the XMPP Council and result in publication of a revised document version (e.g., version 2.1). In general, any such changes are made by the XMPP Extensions Editor or XEP author in consultation with the XMPP Council, discussed within the Standards SIG if appropriate, and agreed upon by the full XMPP Council. Upon agreement regarding the exact changes, the XMPP Council shall instruct the XMPP Extensions Editor to publish a revised version of the XEP and announce the existence of the revised version through the normal channels (e.g., on the XSF website and to the Standards list). Naturally, if members of the XMPP developer community have concerns regarding the changes made, they are free to discuss the matter in the relevant forum (usually the Standards list) before or after the revised version has been published. Ultimate authority for Final and Active XEPs rests with the XMPP Council, which can at its discretion demand the reversal of any changes made by the XMPP Extensions Editor or the XEP author while the XEP is in the Final or Active state.
The process is similar with regard to Historical and Informational XEPs that have achieved a status of Active (corresponding to document version 1.0 of the XEP): the XMPP Council agrees on the exact changes to be made and instructs the XMPP Extensions Editor to publish and announce a revised version (e.g., version 1.1). Here again the XMPP Council bears responsibility for any changes and public discussion is welcome.
Procedural XEPs may be modified more frequently as the XMPP Standards Foundation gains experience with the processes defined therein. For example, XEP-0001 is modified periodically in order to document processes previously not made explicit or to modify existing processes based on experience with the XSF's standards process; similar changes are sometimes made to the XMPP Registrar Function (XEP-0053) [32] XEP and to various SIG-related XEPs. Changes to these XEPs are discussed by the XMPP Council, XSF Board of Directors, XSF membership, and Standards SIG as appropriate, and exact changes are agreed to by the relevant approving body (XMPP Council or XSF Board of Directors). The approving body then instructs the XMPP Extensions Editor to publish and announce the revised version as described above.

11 Expiration Dates

In rare cases, a protocol enhancement may be accepted as an interim solution, especially when it is recognized that expected future improvements in technology or the underlying XMPP protocols will make possible a much better solution to the problem at hand (e.g., a better protocol for user avatars may be contingent upon the development of a robust protocol for publish/subscribe functionality). In such cases, a XEP may be approved provisionally and be assigned an expiration date.
The exact form of such an expiration date shall be left up to the discretion of the XMPP Council. However, the preferred form is to assign an expiration date of six (6) months in the future, at which time the XMPP Council must re-affirm the status of the XEP and, if desired,

[32] XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

extend the expiration date for another six (6) months. Although this process may continue indefinitely (although that is unlikely), it has the virtue of forcing the XMPP Council and XMPP developer community to re-examine the provisional protocol on a fairly regular basis in the light of technological changes. Alternatively, a XEP may be assigned a ”soft” expiration date: that is, the XEP will expire when an expected future protocol comes into existence, whenever that may be. In either case, the status of the XEP shall be changed to Deprecated when it expires.
In addition, an expiration date may be assigned when the status of a XEP is changed from Final (or, potentially, Stable) to Deprecated. In this case, the expiration date applies to the date when the XEP is expected to change from Deprecated to Obsolete. These dates may be flexible; however it is expected that they will follow the same six-month rule as provisional protocol enhancements.

12 Security Considerations

Every XMPP Extension Protocol specification must contain a section entitled ”Security Considerations”, detailing security concerns or features related to the proposal; in particular, a Standards Track XEP should list the security threats that the protocol addresses and does not address, as well as security issues related to implementation of the protocol and deployment of such implementations. XEP authors should refer to RFC 3552 [33] for helpful information about documenting security considerations and should also confer with the XMPP Extensions Editor and/or XMPP Council regarding this important task.

13 IANA Considerations

Some XMPP Extension Protocols may require interaction with the Internet Assigned Numbers Authority (IANA) [34]. The IANA acts as a clearinghouse to assign and coordinate the use of numerous Internet protocol parameters, such as MIME types and port numbers (e.g., the TCP ports 5222, 5269, and 5280 used by the XMPP developer community are registered with the IANA). Whether or not a XEP requires registration of parameters with the IANA, that fact must be noted and explained in a distinct section of the XEP entitled ”IANA Considerations”. Registration with the IANA must not occur until the registration has been approved by the XMPP Council (e.g., by advancement of a XEP to a status of Stable or Active), and must be initiated by the XMPP Registrar in consultation with the XEP author, not by the XEP author directly with the IANA.

[33] RFC 3552: Guidelines for Writing RFC Text on Security Considerations <http://tools.ietf.org/html/rfc3552>.

[34] 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/>.

14 XMPP Registrar Considerations

The XMPP Registrar [35] performs a function similar to the IANA, although limited to the XMPP developer community. It does so by reserving protocol namespaces and by uniquely assigning parameters for use in the context of XMPP protocols (for example, the categories and types used in Service Discovery (XEP-0030) [36]).
Whether or not a XEP requires registration of protocol namespaces or parameters with the XMPP Registrar, that fact must be noted and explained in a distinct section of the XEP entitled "XMPP Registrar Considerations". Such registration should not occur until a XEP advances to a status of Stable (Standards Track XEPs) or Active (Informational and Historical XEPs). Registration of protocol namespaces is initiated by the XMPP Extensions Editor when a XEP advances to Stable or Active. Registration of particular parameters used within a specification may be initiated by a XEP author within the text of the XEP, or by an implementor of the XEP after it has advanced to Stable or Active. For details regarding the XMPP Registrar and its processes, refer to XMPP Registrar Function (XEP-0053) [37].
A XEP may also request that a new registry is to be created by the XMPP Registrar. The XEP author must clearly define the nature of the new registry as well as the process for submitting data to the registry, and should do so in collaboration with the Registrar.

15 XML Schema

XMPP Extension Protocol specifications that define official XSF protocols must include a schema that conforms to XML Schema Part 1 [38] and XML Schema Part 2 [39].
The schema for the XEP format itself is as follows:

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

<!--

Copyright (c) 1999 - 2021 XMPP Standards Foundation

Permission is hereby granted, free of charge, to any person obtaining
    a copy
of this software and associated documentation files (the "Software"),
    to deal
in the Software without restriction, including without limitation the
    rights
```

[35] 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/>.
[36] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[37] XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.
[38] XML Schema Part 1: Structures <http://www.w3.org/TR/xmlschema11-1/>.
[39] XML Schema Part 2: Datatypes <http://www.w3.org/TR/xmlschema11-2/>.

```
to use, copy, modify, merge, publish, distribute, sublicense, and/or
    sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
    included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
    EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
    MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
    SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
    ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
     IN
THE SOFTWARE.

-->

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='http://www.xmpp.org/extensions'
    xmlns='http://www.xmpp.org/extensions'
    elementFormDefault='qualified'>

  <xs:element name='xep'>
    <xs:annotation>
      <xs:documentation>

        This schema defines the document format for XMPP Extension
        Protocols (XEPs). For further information about XEPs, visit:

           http://www.xmpp.org/extensions/

        The canonical URL for this schema is:

           http://www.xmpp.org/extensions/xep.xsd

      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='header'/>
        <xs:element ref='section1' maxOccurs='unbounded'/>
```

```
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='header'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='title' type='xs:string'/>
      <xs:element name='abstract' type='xs:string'/>
      <xs:element ref='legal'/>
      <xs:element name='number' type='xs:byte'/>
      <xs:element ref='status'/>
      <xs:element name='lastcall' minOccurs='0' type='xs:string'/>
      <xs:element name='interim' minOccurs='0' type='empty'/>
      <xs:element ref='type'/>
      <xs:element name='sig' type='xs:string'/>
      <xs:element name='approver' type='xs:string'/>
      <xs:element ref='dependencies'/>
      <xs:element ref='supersedes'/>
      <xs:element ref='supersededby'/>
      <xs:element name='shortname' type='xs:NCName'/>
      <xs:element ref='schemaloc' minOccurs='0' maxOccurs='unbounded
          '/>
      <xs:element name='registry' minOccurs='0' type='empty'/>
      <xs:element name='discuss' minOccurs='0' type='xs:string'/>
      <xs:element name='expires' minOccurs='0' type='xs:string'/>
      <xs:element ref='author' minOccurs='1' maxOccurs='unbounded'/>
      <xs:element ref='revision' minOccurs='1' maxOccurs='unbounded'
          />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='legal'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='copyright' type='markup'/>
      <xs:element name='permissions' type='markup'/>
      <xs:element name='warranty' type='markup'/>
      <xs:element name='liability' type='markup'/>
      <xs:element name='conformance' type='markup'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='status'>
  <xs:simpleType>
    <xs:restriction base='xs:NCName'>
      <xs:enumeration value='Active'/>
```

```
      <xs:enumeration value='Deferred'/>
      <xs:enumeration value='Deprecated'/>
      <xs:enumeration value='Draft'/>
      <xs:enumeration value='Experimental'/>
      <xs:enumeration value='Final'/>
      <xs:enumeration value='Obsolete'/>
      <xs:enumeration value='Proposed'/>
      <xs:enumeration value='ProtoXEP'/>
      <xs:enumeration value='Rejected'/>
      <xs:enumeration value='Retracted'/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name='type'>
  <xs:simpleType>
    <xs:restriction base='xs:string'>
      <xs:enumeration value='Historical'/>
      <xs:enumeration value='Humorous'/>
      <xs:enumeration value='Informational'/>
      <xs:enumeration value='Procedural'/>
      <xs:enumeration value='Standards Track'/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name='dependencies'>
  <xs:complexType>
    <xs:sequence minOccurs='0' maxOccurs='unbounded'>
      <xs:element name='spec' type='xs:string'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='supersedes'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='spec' type='xs:string'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='supersededby'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='spec' type='xs:string'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name='schemaloc'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='ns' type='xs:string' minOccurs='0'/>
      <xs:element name='url' type='xs:string'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='author'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='firstname' type='xs:string'/>
      <xs:element name='surname' type='xs:string'/>
      <xs:element name='authornote' type='empty' minOccurs='0'/>
      <xs:element name='org' type='xs:string' minOccurs='0'/>
      <xs:element name='email' type='xs:string' minOccurs='0'/>
      <xs:element name='jid' type='xs:string' minOccurs='0'/>
      <xs:element name='uri' type='xs:string' minOccurs='0'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='revision'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='version' type='xs:string'/>
      <xs:element name='date' type='xs:dateTime'/>
      <xs:element name='initials' type='xs:NCName'/>
      <xs:element ref='remark'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='remark'>
  <xs:complexType>
    <xs:choice>
      <xs:element ref='p' minOccurs='0' maxOccurs='1'/>
      <xs:element ref='ul' minOccurs='0' maxOccurs='1'/>
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:element name='section1'>
  <xs:complexType>
    <xs:choice maxOccurs='unbounded'>
      <xs:element ref='code' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='div' minOccurs='0' maxOccurs='unbounded'/>
```

```
      <xs:element ref='dl' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='example' minOccurs='0' maxOccurs='unbounded'/
          >
      <xs:element ref='ol' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='p' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='section2' minOccurs='0' maxOccurs='unbounded'
          />
      <xs:element ref='table' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='ul' minOccurs='0' maxOccurs='unbounded'/>
    </xs:choice>
    <xs:attribute name='topic' type='xs:string' use='required'/>
    <xs:attribute name='anchor' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='section2'>
  <xs:complexType>
    <xs:choice maxOccurs='unbounded'>
      <xs:element ref='code' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='div' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='dl' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='example' minOccurs='0' maxOccurs='unbounded'/
          >
      <xs:element ref='ol' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='p' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='section3' minOccurs='0' maxOccurs='unbounded'
          />
      <xs:element ref='table' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='ul' minOccurs='0' maxOccurs='unbounded'/>
    </xs:choice>
    <xs:attribute name='topic' type='xs:string' use='required'/>
    <xs:attribute name='anchor' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='section3'>
  <xs:complexType>
    <xs:choice maxOccurs='unbounded'>
      <xs:element ref='code' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='div' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='dl' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='example' minOccurs='0' maxOccurs='unbounded'/
          >
      <xs:element ref='ol' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='p' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='section4' minOccurs='0' maxOccurs='unbounded'
          />
      <xs:element ref='table' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='ul' minOccurs='0' maxOccurs='unbounded'/>
```

```
    </xs:choice>
    <xs:attribute name='topic' type='xs:string' use='required'/>
    <xs:attribute name='anchor' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='section4'>
  <xs:complexType>
    <xs:choice maxOccurs='unbounded'>
      <xs:element ref='code' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='div' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='dl' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='example' minOccurs='0' maxOccurs='unbounded'/
          >
      <xs:element ref='ol' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='p' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='table' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='ul' minOccurs='0' maxOccurs='unbounded'/>
    </xs:choice>
    <xs:attribute name='topic' type='xs:string' use='required'/>
    <xs:attribute name='anchor' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='div'>
  <xs:complexType>
    <xs:choice maxOccurs='unbounded'>
      <xs:element ref='div' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='p' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='example' minOccurs='0' maxOccurs='unbounded'/
          >
      <xs:element ref='code' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='ul' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='ol' minOccurs='0' maxOccurs='unbounded'/>
    </xs:choice>
    <xs:attribute name='class' type='xs:string' use='optional'/>
    <xs:attribute name='style' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='p' type='markup'/>

<xs:element name='ul'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='li' minOccurs='1' maxOccurs='unbounded'/>
    </xs:sequence>
    <xs:attribute name='class' type='xs:string' use='optional'/>
    <xs:attribute name='style' type='xs:string' use='optional'/>
```

```
  </xs:complexType>
</xs:element>

<xs:element name='ol'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='li' minOccurs='1' maxOccurs='unbounded'/>
    </xs:sequence>
    <xs:attribute name='class' type='xs:string' use='optional'/>
    <xs:attribute name='start' type='xs:byte' use='optional'/>
    <xs:attribute name='style' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='li' type='markup'/>

<xs:element name='dl'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='di' minOccurs='1' maxOccurs='unbounded'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='di'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='dt' minOccurs='1' maxOccurs='1'/>
      <xs:element ref='dd' minOccurs='1' maxOccurs='1'/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='dt' type='xs:string'/>

<xs:element name='dd' type='markup'/>

<xs:element name='img'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='source' use='required'/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name='link'>
  <xs:complexType>
```

```
    <xs:simpleContent>
      <xs:extension base='xs:string'>
        <xs:attribute name='url' use='required'/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name='note' type='markup'/>

<xs:element name='example'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='xs:string'>
        <xs:attribute name='caption' use='optional'/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name='code'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='xs:string'>
        <xs:attribute name='caption' use='optional'/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name='table'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='tr' minOccurs='1' maxOccurs='unbounded'/>
    </xs:sequence>
    <xs:attribute name='caption' type='xs:string' use='optional'/>
  </xs:complexType>
</xs:element>

<xs:element name='tr'>
  <xs:complexType>
    <xs:choice>
      <xs:element ref='th' minOccurs='0' maxOccurs='unbounded'/>
      <xs:element ref='td' minOccurs='0' maxOccurs='unbounded'/>
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:element name='th'>
```

```
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute name='colspan' use='optional'/>
          <xs:attribute name='rowspan' use='optional'/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name='td'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute name='colspan' use='optional'/>
          <xs:attribute name='rowspan' use='optional'/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:complexType name='markup' mixed='true'>
    <xs:choice minOccurs='0' maxOccurs='unbounded'>
      <xs:element name='br' type='empty'/>
      <xs:element name='cite' type='xs:token'/>
      <xs:element name='dfn' type='xs:token'/>
      <xs:element name='em' type='xs:token'/>
      <xs:element ref='img'/>
      <xs:element ref='link'/>
      <xs:element ref='note'/>
      <xs:element name='span' type='xs:token'/>
      <xs:element name='strong' type='xs:token'/>
      <xs:element name='tt' type='xs:token'/>
    </xs:choice>
    <xs:attribute name='class' type='xs:string' use='optional'/>
    <xs:attribute name='style' type='xs:string' use='optional'/>
  </xs:complexType>

  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value=''/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```