This specification defines a formatted text syntax for use in instant messages with simple text styling.
Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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).
## Contents

1. **Introduction** 1
2. **Requirements** 1
3. **Use Cases** 1
4. **Glossary** 2
5. **Business Rules** 2
   5.1 Blocks 2
      5.1.1 Plain 3
      5.1.2 Preformatted Text 3
      5.1.3 Quotations 4
   5.2 Spans 4
      5.2.1 Plain 5
      5.2.2 Emphasis 5
      5.2.3 Strong Emphasis 5
      5.2.4 Strike through 6
      5.2.5 Preformatted Span 6
6. **Implementation Notes** 6
7. **Accessibility Considerations** 7
8. **Security Considerations** 7
9. **IANA Considerations** 7
10. **XMPP Registrar Considerations** 7
11. **XML Schema** 7
12. **Acknowledgements** 8
1 Introduction

Historically, XMPP has had no system for simple text styling. Instead, specifications like XHTML-IM (XEP-0071) that require full layout engines have been used, leading to numerous security issues with implementations. Some entities have also performed their own styling based on identifiers in the body. While this has worked well in the past, it is not interoperable and leads to entities each supporting their own informal styling languages. This specification aims to provide a single, interoperable formatted text syntax that can be used by entities that do not require full layout engines.

2 Requirements

- Clients that do not support this specification MUST still be able to receive messages sent by clients using this specification and display them in a human-readable form.
- Clients that support this specification MUST NOT be required to use a layout engine such as HTML or LaTeX.
- Messages formatted using this specification MUST NOT hinder readability on receiving clients regardless of client background color, contrast, or window size.
- Messages formatted using this specification MUST NOT hinder readability by users with color vision deficiency or impaired vision.
- Messages formatted with this specification MUST render correctly in locales with right-to-left (RTL) layouts without causing confusion.
- Clients that support this specification MUST NOT be required to extract metadata unrelated to formatting or text style from the message.
- Servers MUST NOT need to implement any new functionality for this specification to be supported.

3 Use Cases

- As a user sending an instant message to a friend, I want to be able to emphasize an important part of my message.
- As a software developer, I want to be able to send code as pre-formatted, monospace, block or inline text to another developer.
- As a multi-user chat user I want to add context to my reply by quoting an earlier message in the chat.

4 Glossary

Many important terms used in this document are defined in Unicode \(^2\). The terms “left-to-right” (LTR) and “right-to-left” (RTL) are defined in Unicode Standard Annex #9 \(^3\). The term “formatted text” is defined in RFC 7764 \(^4\).

**Block** Any chunk of text that can be parsed unambiguously in one pass. Blocks may contain one or more children which may be other blocks or spans. For example:

- A single line of text comprising one or more spans
- A block quotation
- A preformatted code block

**Formal markup language** A structured markup language such as LaTeX, SGML, HTML, or XML that is formally defined and may include metadata unrelated to formatting or text style.

**Plain text** Text that does not convey any particular formatting or interpretation of the text by computer programs.

**Span** A group of text that may be rendered inline alongside other spans. Spans may be either plain text with no formatting applied, or may be formatted text that is enclosed by two styling directives. Spans are always children of blocks and may not escape from their containing block. Some spans may contain child spans. The following all contain spans marked by parenthesis:

- (plain span) (*strong span*) (_emphasized span_) (_emphasized span containing (*strong span*)_) (span one)(*span two*)

**Styling directive** A character or set of characters that indicates the beginning of a span or block. For example, in certain contexts the characters '*' (U+002A ASTERISK), and '^-' (U+005F LOW LINE) may be styling directives that indicate the beginning of a strong or emphasis span and the string ``'''' (U+0060 GRAVE ACCENT) may be a styling directive that indicate the beginning of a preformatted code block.

**Whitespace character** Any Unicode scalar value which has the property ”White_Space” or is in category Z in the Unicode Character Database.

5 Business Rules

5.1 Blocks

Parsers implementing message styling will first parse blocks and then parse child blocks or spans if allowed by the specific block type.

---

\(^1\)The Unicode Standard, The Unicode Consortium <http://www.unicode.org/versions/latest/>.


5.1.1 Plain

Individual lines of text that are not inside of a preformatted text block are considered a "plain" block. Plain blocks are not bound by styling directives and do not imply formatting themselves, but they may contain spans which imply formatting. Plain blocks may not contain child blocks.

Listing 1: Plain block text

```html
<body>
  (There are three blocks in this body marked by parens,)
  (but there is no * formatting)
  (as spans* may not escape blocks.)
</body>
```

5.1.2 Preformatted Text

A preformatted text block is started by a line beginning with "'\n'" (U+0060 GRAVE ACCENT), and ended by a line containing only three grave accents or the end of the parent block (whichever comes first). Preformatted text blocks cannot contain child blocks or spans. Text inside a preformatted block SHOULD be displayed in a monospace font.

Listing 2: Preformatted block text

```html
<body>
  '{}' ' ignored
  (println "Hello, world!"
  '()()'
  This should show up as monospace, preformatted text
</body>
```

Listing 3: No closing preformatted text sequence

```html
<body>
  &gt; '{}'
  &gt; (println "Hello, world!"
  The entire blockquote is a preformatted text block, but this line
  is plaintext!
</body>
```
5.1.3 Quotations

A quotation is indicated by one or more lines with a byte stream beginning with a ‘>’ (U+003E GREATER–THAN SIGN). They are terminated by the first new line that is not followed by a greater-than sign, or the end of the parent block (whichever comes first). Block quotes may contain any child block, including other quotations. Lines inside the block quote MUST have the first leading whitespace character trimmed before parsing the child block. It is RECOMMENDED that text inside of a block quote be indented or distinguished from the surrounding text in some other way.

Listing 4: Quotation (LTR)

```html
<body>
 &gt; That that is, is.

 Said the old hermit of Prague.
</body>
```

Listing 5: Nested Quotation

```html
<body>
 &gt;&gt; That that is, is.
 &gt; Said the old hermit of Prague.

 Who?
</body>
```

5.2 Spans

Matches of spans between two styling directives MUST contain some text between the two styling directives and the opening styling directive MUST be located at the beginning of the line, after a whitespace character, or after a different opening styling directive. The opening styling directive MUST NOT be followed by a whitespace character and the closing styling directive MUST NOT be preceeded by a whitespace character. Spans are always parsed from the beginning of the byte stream to the end and are lazily matched. Characters that would be styling directives but do not follow these rules are not considered when matching and thus may be present between two other styling directives.

For example, each of the following would be styled as indicated:

- *strong*
- plain *strong* plain
- *strong* plain *strong*
- *strong*plain*
5 BUSINESS RULES

• * plain *strong*

Nothing would be styled in the following messages (where \"\n\" represents a new line):

• not strong*
• *not strong
• *not \n strong*
• *not *strong
• **
• ****

5.2.1 Plain

Any text inside of a block that is not part of another span is implicitly considered to be inside of a "plain text" span.

Listing 6: Plain

```html
<body>
  (Two spans, both )(*alike in dignity*)
</body>
```

5.2.2 Emphasis

Text enclosed by _ (U+005F LOW LINE) is emphasized and SHOULD be displayed in italics.

Listing 7: Italic

```html
<body>
  The full title is _Twelfth Night, or What You Will_ but _most_ people shorten it.
</body>
```

5.2.3 Strong Emphasis

Text enclosed by * (U+002A ASTERISK) is strongly emphasized and SHOULD be displayed with a heavier font weight than the surrounding text (bold).
The full title is "Twelfth Night, or What You Will" but *most* people shorten it.

5.2.4 Strike through

Text enclosed by '~' (U+007E TILDE) SHOULD be displayed with a horizontal line through the middle.

Everyone ~dis~likes cake.

5.2.5 Preformatted Span

Text enclosed by a `'' (U+0060 GRAVE ACCENT) is a preformatted span SHOULD be displayed inline in a monospace font. A preformatted span may only contain a single plain span. Inline formatting directives inside the preformatted span are not rendered. For example, the following all contain valid preformatted spans:

- This is 'monospace'
- This is "monospace"
- This is **monospace and bold**

6 Implementation Notes

This document does not define a regular grammar and thus styling cannot be matched by a regular expression. Instead, a simple parser can be constructed by first parsing all text into blocks and then recursively parsing the child-blocks inside block quotations, the spans inside individual lines, and by returning the text inside preformatted blocks without modification. It is RECOMMENDED that formatting characters be displayed and formatted in the same
manner as the text they apply to. For example, the string "*emphasis*" would be rendered as "*emphasis*".

7 Accessibility Considerations

When displaying text with formatting, developers should take care to ensure sufficient contrast exists between styled and unstyled text so that users with vision deficiencies are able to distinguish between the two. Formatted text may also be rendered poorly by screen readers. When applying formatting it may be desirable to include directives to exclude formatting characters from being read.

8 Security Considerations

Authors of message styling parsers should take care that improperly formatted messages cannot lead to buffer overruns or code execution.

9 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) 5.

10 XMPP Registrar Considerations

This specification requires no interaction with the XMPP Registrar 6.

11 XML Schema

This document does not define any new XML structure requiring a schema.

---

5The 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/>.

6The 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/>.
12 Acknowledgements

The author wishes to thank Kevin Smith for his review and feedback.