RFC Data Visualizations Accessibility Guidance in HTML Documents

Internet Engineering Task Force (IETF)

August 13, 2024

Table of Contents

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**

1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Introduction

This document details some possible approaches and techniques for evaluating various types of data visualization assets, with a view to incorporating accurate and accessible visual descriptions for them.

We've included guidance on how to provide visual descriptions for various types of diagrams in HTML documents and several examples to illustrate the most suitable methods of making their data and context clear. Note that the visual descriptions provided in these examples are solely based on our understanding of the diagrams within their current context, which may differ from the author's intent, and we expect that adjustments will be made to these descriptions to ensure they are fully accurate and relevant.

In many cases the suggested long description is presented inline in this document, and it is assumed that the sampled Microsoft Word formatting for headings, lists and tables will translate into proper semantic HTML elements that fit within the overall structure of a corresponding RFC document. Additionally, the report is accompanied by an HTML document with examples illustrating various methods we recommend for SVGs which could not be rendered in Microsoft Word. The file is located in the "SVG Examples" folder, along with a stylesheet and some JavaScript utilities.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Glossary

Visual Description

A visual description is a textual representation of static media or objects, most commonly intended for those who may not be able to see the media or objects, but can benefit all people. There are two general types of visual description:

- Short Descriptions; and
- Long Descriptions.

Short Description

A short description provides a very brief and general sense of the content of an image or object. The length of a short description should be up to 60 words max.

Short descriptions are most commonly used on the web or other electronic formats and are mapped to alt text fields.

Note: Alt text is a commonly misused term derived from the term "alternative text" that means a short visual description.

Long Description

A long description should comprehensively describe the most important elements of an image augmented by a few prioritized details. The length can range from one to multiple paragraphs depending on the complexity of the image and the context.

Long descriptions must be surfaced accessibly providing maximum user customization via an appropriate bespoke solution. They should also be surfaced visually so that all visitors can access them.

Refer to: Inclusive Design Glossary.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Purpose

Diagrams must have informative visual descriptions that accurately convey the purpose of the image, whether it is to visualize content explained in adjacent paragraphs, summarize information, or to emphasize a specific point of view.

The starting point is to determine what a diagram is trying to convey. We should only look into practical methods of making the same information available to all users once this aspect has been unambiguously established.

While various methods are available to surface this information, the aim is to find the most efficient way of equitably presenting the diagram content to all users. Firstly, it's important to point out that all documents must have an HTML version. This facilitates the provision of visual descriptions detailed in this document, and allows for the structural markup required to accurately convey the info and relationships which are visually available to users of access technology.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Approach

In the provided HTML documents, diagrams are implemented in various ways, including <svg> elements, and preformatted content via the <pre> element. Different methods offer different levels of accessibility and require distinct methods of supplying a visual description.

Data Visualization Techniques

Scalable Vector Graphics (SVGs)

Scalable Vector Graphics (SVGs) are a versatile image format used extensively in web development. Unlike traditional bitmap images, SVGs are vector-based, meaning they can scale infinitely without losing quality. This makes them ideal for creating responsive, high-resolution graphics that look sharp on any device.

SVGs offer several key accessibility advantages:

- **Scalability**: SVGs maintain clarity and detail at any size, ensuring that images are always sharp and readable, regardless of the screen resolution or zoom level.
- **Text-based Content**: If the XML within SVGs is structured appropriately, the text within them will be selectable, searchable, and readable by access technology like screen readers.
- **Interactive and Animatable**: SVGs can be enhanced with CSS and JavaScript, allowing for dynamic and interactive graphics that can improve user engagement and provide additional context.
- **ARIA Support**: SVGs can incorporate ARIA (Accessible Rich Internet Applications) attributes to offer more semantic information to access technologies, making it easier for users with disabilities to understand and navigate the content.
- **Efficiency**: SVGs often have smaller file sizes compared to bitmap images, leading to quicker load times and a better experience for users with limited bandwidth.

SVGs offer great flexibility when providing accessible content. Visual descriptions can be made available to screen reader users in two ways:

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- A short description can be added to the SVG itself, and a visible long description can be provided adjacent to the SVG as needed. This is the preferred path for most non-complex diagrams and the most efficient method.
- Alternatively, SVGs can be made fully accessible through semantic markup within the SVG file itself, eliminating the need for additional explanatory text and ensuring that all users have access to the same information seamlessly. However, this process requires extensive knowledge of ARIA roles and attributes, as well as the context being conveyed and its reason for inclusion. For an example, refer to SVG with Built-In Semantic Markup and Visually-Hidden Text later in this document, and the accompanying "SVG Examples" folder.

Preformatted Content

Many diagrams (and even full RFC documents in some cases) are provided as preformatted text using the <pre> element. Preformatted text is displayed in a fixed-width font, preserving both spaces and line breaks exactly as they are in the HTML source. This type of text is often used for displaying code snippets, ASCII art, and structured documents that require specific formatting to be readable.

It is crucial to avoid using preformatted text for full documents, because it severely hinders accessibility. For example:

- Screen readers interpret the content of <pre> elements verbatim, presenting every space and line break in speech and braille. This can be overwhelming and confusing, and causes any custom readability-related settings configured in the software (such as line length) to be ignored.
- Such documents lack any structural markup.
    - Text styling is used to visually structure the document, but programmatic headings are not present. This makes the document nearly impossible to navigate and understand for screen reader users, or people relying on outline tools in their browser.
    - Similarly, content which is formatted as tabular data is not implemented as an actual table, preventing users from exploring it in a meaningful and logical way. For example, all data is communicated without reference to the visible column and/or row headers, and screen reader commands for moving by row and column will be unavailable.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- The fixed-width of preformatted content causes significant problems for screen magnification users, who may need to scroll in both dimensions to read the text when it doesn't scale or reflow.

Here are some example RFCs with no semantic structure, because of each page being wrapped in a <pre> element:

- Example with Header Diagram: https://www.rfc-editor.org/rfc/rfc8200
- Example with Ladder Diagram: https://datatracker.ietf.org/doc/html/draft-loreto-httpbis-trusted-proxy20-01
- Examples of UML diagrams:
  - https://www.rfc-editor.org/rfc/rfc7460
  - https://www.rfc-editor.org/rfc/rfc7143

We have included long description examples for diagrams in such documents, but it is imperative that proper semantic elements be used for the HTML documents, along with CSS for styling. Only then should the focus be on providing visual descriptions for diagrams, because the documents are currently inaccessible in many other, more impactful ways.

Bitmap Images (Raster Images)

Bitmap images, also known as raster images, are a type of digital graphic composed of a grid of individual pixels, each with its own color value. These pixels collectively form an image, with the quality and detail determined by the number of pixels (resolution).

The main accessibility concern with using bitmap images is scalability: when magnified, images can appear pixelated and hence be hard or impossible to understand. It is therefore best to avoid using bitmap images for diagrams altogether, because they are not scalable, and don't offer any of the other, previously noted advantages exhibited by SVGs (including retention of clarity when zoom is applied).

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Visual Description

All diagrams must have a short description, and/or an accessible long visual description using semantic markup, which not only emphasizes the purpose of the diagram but is also easy to navigate and explore by all users. Whenever a short description is provided, careful examination is needed to determine if this is sufficient, or whether the diagram must be accompanied by a structured long description within the document.

This means that:

- For certain types of images/diagrams, a short description implemented as a text alternative is sufficient.
- For other types, a short description is necessary but not sufficient. In these cases:
  - We need to use semantic markup, such as headings, lists, tables, etc. to properly explain the information conveyed in the diagram in the document itself. The short description for the diagram can be brief and point to this structured information.
  - Alternatively, the markup may be built into the diagram itself using ARIA roles and/or semantic HTML elements depending on the data visualization technique.

Short Description

When a diagram with a short description is encountered, screen reader users must be informed of the following:

- the presence of an image;
- the relationship between the image and the caption text;
- the short description which briefly indicates the diagram's purpose and description; and
- the location of a long description when applicable.

The short description must not duplicate the caption text in <figcaption>, for example:

- the caption may indicate the name of the illustrated component or process; while

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- the short description can summarize the diagram's purpose, design intent, and key elements, along with the location of a long description, if needed.

Short descriptions can be added using different techniques depending on the diagram's implementation.

SVGs

The short description is constructed within each SVG using the following adjustments:

- Give the <svg> the role of "img" so that it is consistently identified as a "graphic" element across all browsers.
- Add a <title> element inside the <svg> with brief, informative alt text.
  - This should present the purpose of the diagram.
  - Some form of punctuation should exist such as a colon and a space at the end of the text so that it is separated from the text in the <desc> element in the screen reader speech output.
- Apply a unique id attribute to the <title>.
- Add a <desc> element inside the <svg> with a description of the diagram.
  - If the description requires structure, simply point to where the information is presented in the text, e.g. "Refer to the following list of decryption steps." or similar. This means that the structured long description must be present in text inside the document as referenced.
  - Keep the description as simple as possible.
- Apply a unique id attribute to the <desc>.
- Apply the IDs of the <title> and <desc> as the value of an aria-labelledby attribute on the <svg> itself, with the two separated by a space character.

For example:

```
<figure>
      <svg role="img" aria-labelledby="ID_title ID_desc" ...>
            <title id="ID_title">An informative brief title</title>
            <desc id="ID_desc">Additional information about the diagram and/or the location of the long description</desc>
```


312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

```
    </svg>
    <figcaption>The visible caption text</figcaption>
</figure>
```

Preformatted Content

As previously mentioned, preformatted text is problematic for users of access technology.

For diagrams presented as preformatted text (<pre>), there are three accessible alternatives:

- Convert the preformatted text into regular HTML with appropriate CSS to maintain the visual formatting and use semantic HTML elements, such as headings and lists, to structure the diagram components for screen reader users. This method has the advantage of not needing to add additional content, such as a long description, to make diagrams accessible.
- Apply role="img" to the <pre> element and provide a short description using the aria-label or aria-labelledby attributes. Additionally, include a detailed long description in a structured format, below the diagram that conveys the same information. Note that this approach does not apply to entire documents wrapped in a <pre> tag.
- Convert the diagrams into <svg> elements. Depending on the complexity of the diagram, the necessary information can be embedded within the SVG, or a short description can be provided along with a structured long description below the SVG. This method offers the most flexibility and potential for creating fully accessible diagrams, but will be undermined if the rest of the document is presented as preformatted text.

Bitmap Images (Raster Images)

These are best avoided for diagrams because they are not scalable. Although none of the provided examples included diagrams which were rendered using the <img> element, there may be images in some RFC documents which do use it. In such cases, a short description must be added as the value of the alt attribute on the <img> element, with the long description advice elsewhere in this document followed as required.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Long Description

Regardless of the data visualization technique, e.g. <svg>, <pre>, etc. diagrams must almost always be accompanied by long descriptions presented in the document as visible text.

Due to the technical nature of RFC documents, it is not recommended to use a paragraph-based format (prose) for the long description. Structured information will often be easier to comprehend and navigate when compared to the same content in paragraph blocks. Long descriptions can be presented in the form of one or more lists (including nested lists where appropriate), data tables, sections structured with headings, and other semantic markup, presented using the following methods:

- inside the document either before or after the diagram;
- inside a modal dialog (See: Appendix: Dialog); or
- using a disclosure component (See: Appendix: Disclosure).

Each method has its own advantages and disadvantages, and it may be that this decision should be made on a case-by-case basis. Overall though, presenting the long description inside a modal dialog is the recommended path for several reasons:

- It only requires the addition of a button following the diagram to open the dialog, which will not noticeably increase the length of the document.
- The HTML native <dialog> element has good support and comes with default keyboard focus handling.
- Inside the dialog, the first heading should be marked up as <h1>, leaving authors with five heading levels from <h2> to <h6> to structure the visual description.

Headings

Headings are the preferred tool for screen reader users to explore digital content, understand its structure, and find the desired information. Headings are very useful in structuring the long description and emphasizing essential information.

Headings may precede lists or tables which present the content illustrated visually in the diagram in an accessible and navigable format. Additional headings can be used to further break down the long description where appropriate.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

When rendering heading markup, ensure that heading levels fit within the document structure and that no levels are skipped.

Examples

- Finite State Machines (FSMs)
- Call Flow/ Ladder Diagrams
- Other Diagrams: Byte Layout with Headings and Table in Dialog as Long Description

Lists

Whenever a diagram illustrates a flow, or a process which involves several steps, these should be listed within one or more ordered and/or unordered lists, with valid nesting where necessary. The list should be preceded by a heading which can be referenced in the short description when pointing to the location of the long description.

Examples

- Finite State Machines (FSMs)
- Call Flow/ Ladder Diagrams
- Unified Modeling Language (UML) Diagrams
- Other Diagrams: SVG with Inline List as Long Description in Disclosure

Tables

Tables are extremely useful when data must be understood within a specific context because they allow us to make visual associations programmatically available.

Screen reader users can navigate tables in many ways, including by column or by row, enabling them to extract information without the need to mentally filter it out of a long paragraph or list item. Correctly marked up column and row headers help with this by linking corresponding context/meaning with each data point.

Here is some guidance on the required table markup:

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- Use the <table> element to encompass the component.
- To provide a table with a unique and informative accessible name, use the <caption> element inside the table, apply aria-label to the <table> directly, or use aria-labelledby (also on the <table> element) pointing to the ID of a relevant corresponding heading.
- Each row must be wrapped in <tr>.
- Column and row headers must be marked up as <th>.
- Data cells must be <td>.

For further guidance refer to: MDM Web Docs: HTML table Element.

Scenarios where tables can offer a useful visual description in an alternate format include:

- illustrate byte structure/layouts;
- present the data from a Venn Diagram or chart; or
- indicate transitions to a new state based on input, e.g. for Finite State Machines (FSMs).

Examples

- Finite State Machines (FSMs)
- Venn Diagrams
- Header Diagrams
- Other Diagrams: Byte Layout with Inline Table as Long Description
- Other Diagrams: Byte Layout with Headings and Table in Dialog as Long Description

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Other Considerations

In addition to providing an accessible visual description, the accompanying text itself needs to be reviewed to make sure it does not rely on sensory characteristics when referencing the diagram. For example:

- In the Secure Frame (SFrame) document, one paragraph has the text "Arrows indicate two different ways that SFrame protection could be integrated into this media stack". Such a reference will be confusing for users who cannot see the diagram.
- Instead, the paragraph can be rephrased to describe the meaning of the arrows in the diagram. Specifically:
    - The original paragraph is: "For example, Figure 1 shows a typical media sender stack that takes media from some source, encodes it into frames, divides those frames into media packets, and then sends those payloads in SRTP packets. The receiver stack performs the reverse operations, reassembling frames from SRTP packets and decoding. Arrows indicate two different ways that SFrame protection could be integrated into this media stack, to encrypt whole frames or individual media packets.".
    - A rephrased version could be: "For example, Figure 1 shows a typical media sender stack that takes media from some source, encodes it into frames, divides those frames into media packets, and then sends those payloads in SRTP packets. The receiver stack performs the reverse operations, reassembling frames from SRTP packets and decoding. SFrame protection is integrated into this media stack, to encrypt whole frames or individual media packets, in both the encoding and decoding process.".
    - Note: This example needs to be evaluated for technical accuracy.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Advice on Specific Diagram Types

This section aims to provide guidance for specific types of diagrams in shared RFC documents. For most categories we included one or more examples to illustrate the most suitable method for rendering such content and providing visual descriptions.

Mathematical Equations

Example: https://www.rfc-editor.org/rfc/rfc9043.html#figure-4

Use MathML for mathematical content, which allows browsers to translate the markup language into properly formatted equations. MathML ensures that such content is consistently rendered across different browsers and platforms, enhancing accessibility for users who rely on various devices and software. For example:

- Screen magnification users can enlarge the equation while maintaining its integrity and the clear visual distinction of each character.
- MathML content is communicated in an intelligent and meaningful manner for screen reader users who can further explore the different parts of the equation as desired. For a demo on reviewing a sample MathML equation using the NVDA screen reader, refer to the following YouTube video: NVDA and Firefox Reading MathML on Wikipedia.
- MathML is designed to work seamlessly with access technologies, such as braille displays and speech recognition software, providing a more inclusive experience for all users.
- Neurodivergent users or people with reading disabilities can customize the formatting of text without altering the meaning of the equation.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Finite State Machines (FSMs)

An FSM is a collection of states with transitions that occur between these states based on the received input. Refer to: Wikipedia: Finite-state machine.

Long Description with Headings and Lists

One straight forward and highly accessible way of presenting an FSM is to map each state to a heading with a list of rules proceeding this heading.

Here is a structured textual representation of a simple FSM which simply consumes each digit of a binary string and determines whether there are at least two 1s in the string, e.g. expressed by the regular expression ".*11.*". Q0 through Q2 are used to name the states as this is customary in finite automata, but these states can be called anything, including human-readable names. Epsilon refers to the empty string (i.e. the end of the input data). Instead of circles and arrows, we are using headings and list items.

FSM Sample

Q0

- 0 goes to Q0
- 1 goes to Q1
- Epsilon returns false

Q1

- 0 goes to Q0
- 1 goes to Q2
- Epsilon returns false

Q2

- 0 goes to Q0
- 1 goes to Q2

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- Epsilon returns true

FSM Example

URL: https://www.rfc-editor.org/rfc/rfc3261#section-17.1.1.2

Here is another practical example where the long description is rendered using headings and lists. A short description should be added to the <pre> element wrapping the diagram (if this is the preferred data visualization technique) as previously instructed. See: Short Description: Preformatted Content. Specifically:

- apply role="img" to the <pre> element; and
- provide a short description, e.g. "The four states and transitions as structured in the following section." using the aria-label attribute on the <pre> element.

The figure with the diagram and the caption must all use semantic HTML markup using <figure> and <figcaption>. Then the component should be followed by the long description using headings and lists, as follows:

1. Calling:

- Initial state when an INVITE request is sent from the Transaction User (TU).
- Timer A is started, and if it fires, another INVITE is sent.
- If Timer B fires or a transport error occurs, the TU is informed.

2. Proceeding:

- Entered when a provisional response (1xx) is received.
- If a final response (2xx or 300-699) is received, the state transitions to Completed.
- Provisional responses (1xx) are forwarded to the TU.

3. Completed:

- Entered when a final response (300-699) is received.
- An ACK is sent in response to the final response.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- Timer D is started to handle retransmissions and ensure reliability.
- If Timer D fires or a transport error occurs, the state transitions to Terminated.

4. Terminated:

- Final state indicating that the transaction is completed.
- This state can be reached from Calling if a 2xx response is received and forwarded to the TU.
- This state is also reached from Completed if Timer D fires or a transport error occurs.

Long Description as Transition Table

Another way of representing an FSM is a transition table. For each state, each input causes a transition to a new state. These transitions can either flow to the same state such as in the case of receiving 0 on Q0, 0 on Q1, or 0 or 1 on Q2. They can also flow to a different state such as receiving 1 on Q0 or 1 on Q1.

Sample Transition Table

Input	Q0	Q1	Q2
0	Q0	Q0	Q2
1	Q1	Q2	Q2

FSM Example (Table)

URL: https://www.rfc-editor.org/rfc/rfc6733.html#section-5.6

In this example, the diagram is already followed by a visual description in a table layout, with "state", "event", "action", "next", and "state" as column headers, and each state serving as a row header. The table representation is very useful visually as it allows users to easily scan the available states and establish the available information and relationships. It is imperative that native table markup is used so that users of access technology can equally understand and accurately interpret the data in the table.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Currently, the table is formatted using the <pre> element meaning that it is not accessible for screen reader users. See: Tables.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Venn Diagrams

Venn diagrams are a visual way of showing percentage overlap between 2 or more groups about a choice.

Short alt text can convey the nature of the visualization. For example, "A Venn diagram depicting ice cream flavor preferences amongst children and adults.".

Text representing the data must then follow the diagram. This can be in a list format, such as:

- Vanilla: 30% of kids and 60% of adults; and
- Chocolate: 70% of kids and 40% of adults.

If we extend the above to encompass those who like both flavors, i.e. a Venn diagram with 4 circles: one circle for each of kids, adults, chocolate, and vanilla, then a table is more useful.

Group	Chocolate	Vanilla	Both
Kids	50	30	20
Adults	20	50	30

Sample diagrams for ice cream flavor preferences.

Venn Diagram Example

URL: https://www.rfc-editor.org/rfc/rfc8989.html#figure-1

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

A short description should be added to the <svg> as previously instructed. See: Short Description: SVGs. The short description can be constructed in the following format: "A Venn diagram illustrating <purpose of the diagram> with values represented in the following data table".

The figure can then be followed by the data table either inline in the document or as part of a disclosure component or inside a modal dialog. A possible representation of the Venn diagram in Figure 1 in a data table format is shown below:

Group	No overlap	Overlap with People eligible via Path 1, 3 of 5 meetings	Overlap with People eligible via Path 2 or Path 3	Overlap with 2020 actual volunteers	Overlap with all	Total
People eligible via Path 1, 3 of 5 meetings	379	N/A	332	29	102	842
People eligible via Path 2 or Path 3: 1541	1104	332	N/A	3	102	1541
2020 actual volunteers: 135	1	29	3	N/A	102	135

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Header Diagrams

Header diagrams represent the structure of data packets, such as those used in networking protocols. To make these diagrams accessible, a table can be used to indicate the number of bits for each field of the header, along with a description of each field (if available). This table should have column headers, along with using the field name as a row header.

In some cases, e.g. Figure 1: TCP Header Format, the long description is structured using a description list (<dl>). Although the semantic markup is correct and accurately structures the information, they are harder to navigate using a screen reader compared to having the same content presented in a table format.

Header Diagram Example

URL: https://www.rfc-editor.org/rfc/rfc8200#section-3

Once the HTML document is updated to use proper semantic elements such as headings, lists, tables, figures with captions, etc., provide a short description for the diagram, e.g. "The IPv6 header structure, detailed in the following data table.". Use the appropriate method depending on the data visualization technique. See: Short Description.

The data following the diagram is already visually formatted as a data table. It must be implemented using an actual table with column and row headers, for example:

Field Name	Bits	Description
Version	4	Internet Protocol version number
Traffic Class	8	
Flow Label	20	
Payload Length	16	unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets.
Next Header	8	Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Hop Limit	8	Decremented by 1 by each node that forwards the packet. When forwarding, the packet is discarded if Hop Limit was zero when received or is decremented to zero. A node that is the destination of a packet should not discard a packet with Hop Limit equal to zero; it should process the packet normally.
Source Address	128	address of the originator of the packet.
Destination Address	128	address of the intended recipient of the packet (possibly not the ultimate recipient if a Routing header is present).

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Call Flow/ Ladder Diagrams

Ladder diagrams are often used to represent interactions and processes over time. To make these diagrams accessible to blind users, a textual description that outlines each step and interaction in sequence is necessary. This description should detail the entities involved, the actions taken, and the sequence in which these actions occur.

The goal is to convey the same chronological flow and interactions as the visual diagram through clear, structured text.

Steps to Provide an Alternate Description

- Clearly state all the entities (e.g. user-agent, proxy, server) involved in the interaction at the beginning of the description.
- Use numbered list items to describe each step in the sequence. Each list item should correspond to a message in the diagram.
- For each message, clearly state the sender and the receiver. For example, "The user-agent sends a ClientHello message to the proxy.".
- Describe the content or purpose of each message. For example, "The ServerHello message includes the server's certificate and chosen cipher suite.".
- Clearly indicate the direction of each message (e.g. "sends to," "receives from") to help users understand the flow of communication.
- If there are concurrent or parallel messages, describe them separately to avoid confusion.
- Mention any final states or outcomes after the sequence of interactions. For example, "The user-agent and server have successfully established a secure connection.".
- Include any additional information displayed in the diagram that is necessary to understand the sequence, such as the protocols used, or any assumptions made.

Ladder Diagram Example

URL: https://datatracker.ietf.org/doc/html/draft-loreto-httpbis-trusted-proxy20-01#section-4.1

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Once the HTML document is updated to use proper semantic elements, provide a short description for the diagram to indicate its purpose and the location of the long description.

- Use the appropriate method depending on the data visualization technique. See: Short Description.
- The following structured long description can be presented inline in the document or as part of a disclosure component or inside a modal dialog.

Entities Involved:

- User-Agent
- Proxy
- Server

Sequence of Interactions:

1. The User-Agent sends a TLS ClientHello message to the Proxy.
2. The Proxy responds with a ServerHello message to the User-Agent.
3. The User-Agent sends a ChangeCipherSpec message to the Proxy.
4. The Proxy responds with a ChangeCipherSpec message to the User-Agent.
5. The User-Agent sends an HTTP2 GET request (stream X) to the Proxy.
6. The Proxy sends a TLS ClientHello message to the Server.
7. The Server responds with a TLS ServerHello message to the Proxy.
8. The Proxy sends a ChangeCipherSpec message to the Server.
9. The Server responds with a ChangeCipherSpec message to the Proxy.
10. The Proxy sends an HTTP2 GET request (stream Z) to the Server.
11. The Server responds with an HTTP2 200 OK response (stream Z) to the Proxy.
12. The Proxy sends an HTTP2 200 OK response (stream X) to the User-Agent.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Final Conditions:

A secure connection has been established, and HTTP2 requests and responses have been successfully exchanged between the User-Agent and the Server through the Proxy.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Unified Modeling Language (UML) Diagrams

Unified modeling language (UML) is a general-purpose visual modeling language that is intended to provide a standard way to visualize the design of a system. Refer to: Wikipedia: Unified Modeling Language.

A common approach to providing an accessible alternative is to use nested lists and heading mark-up. Each entity in the UML diagram is represented as a level 1 list item additionally implemented as a heading at an appropriate level to fit the document's structure. This method breaks down the visual components of the diagram into a structured, textual format, allowing screen readers to convey the information effectively. Additionally, the heading markup facilitates quick navigation between the different entities.

For each entity, lists can be used to represent its attributes and relationships in a hierarchical manner. This approach ensures that all users can understand the structure and connections within the diagram, as well as the properties of each component.

Steps to Provide a Long Description

- Identify all the entities (e.g. classes, objects, nodes) involved in the UML diagram at the beginning of the description.
- Use lists to detail each entity’s attributes and relationships, making the description easy to follow.
- Clearly describe the relationships between the entities. Specify the type of relationship (e.g. association, inheritance, aggregation, composition) and the direction if applicable.
- Indicate the cardinality (e.g. one-to-many, one-to-one) and multiplicity for relationships if they are depicted in the diagram.
- Describe any important annotations or constraints shown in the diagram, such as stereotypes, notes, or conditions.

While Microsoft Word is limited in that the list and heading semantics cannot be combined, such limitations do not exist in HTML. For example, let's assume that the

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

long description is presented inside a modal dialog. The markup could be structured as indicated in the following code sample:

```
<h1>Sample Long Description for UML Diagram</h1>
<ol>
    <li>
        <h2>First Entity</h2>
        <p>Brief description</p>
        <ul>
            <li>
                <h3>Attributes</h3>
                <ul>
                    <li>Attribute 1</li>
                    <li>Attribute 2</li>
                    <li>Attribute 3</li>
                </ul>
            </li>
            <li>
                <h3>Relationship</h3>
                <ul>
                    <li>Relationship 1, e.g. Has 0 to 1: Entity 2</li>
                    <li>Relationship 1, e.g. Has 1 to many: Entity 3 </li>
                </ul>
            </li>
        </ul>
    </li>
    <li>
        <h2>Second Entity</h2>
        <p>Brief description as needed.</p>
        <ul>
            <li>
                <h3>Attributes</h3>
                <ul>
                    <li>Attribute 1</li>
                    <li>Attribute 2</li>
                    <li>Attribute 3</li>
                </ul>
            </li>
```


312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

```
                <li>
                    <h3>Relationship</h3>
                    <ul>
                        <li>Relationship 1</li>
                        <li>Relationship 2</li>
                    </ul>
                </li>
            </ul>
        </li>
        <li>
            <h2>Third Entity</h2>
            <p>Brief description as needed.</p>
            <ul>
                <li>
                    <h3>Attributes</h3>
                    <ul>
                        <li>Attribute 1</li>
                        <li>Attribute 2</li>
                        <li>Attribute 3</li>
                    </ul>
                </li>
                <li>
                    <h3>Relationship</h3>
                    <ul>
                        <li>Relationship 1</li>
                        <li>Relationship 2</li>
                    </ul>
                </li>
            </ul>
        </li>
</ol>
```

UML Example 1

URL: https://www.rfc-editor.org/rfc/rfc7460#section-5.1.3

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Once the HTML document is updated to use proper semantic elements, provide a short description for the diagram to indicate its purpose and the location of the long description, e.g. "A diagram laying out the structure of energy units and power table.".

- Use the appropriate method depending on the data visualization technique. See: Short Description.
- The following structured long description can be presented inline in the document, as part of a disclosure component, or inside a modal dialog.

Here is the long description formatted using headings and lists:

1. Meter Capabilities

Attribute

- eoMeterCapability

2. Energy Object ID (*)

Attributes

- entPhysicalIndex
- entPhysicalClass
- entPhysicalName
- entPhysicalUUID

Relationships

- Contains the Power Table entity

3. Power Table

Attributes

- eoPower

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- eoPowerNamePlate
- eoPowerUnitMultiplier
- eoPowerAccuracy
- eoPowerMeasurementCaliber
- eoPowerCurrentType
- eoPowerMeasurementLocal
- eoPowerAdminState
- eoPowerOperState
- eoPowerStateEnterReason

UML Example 2

URL: https://www.rfc-editor.org/rfc/rfc7143#section-4.5

Once the HTML document is updated to use proper semantic elements, provide a short description for the diagram to indicate its purpose and the location of the long description.

- Use the appropriate method depending on the data visualization technique. See: Short Description.
- The following structured long description can be presented inline in the document or as part of a disclosure component or inside a modal dialog.

Here is the long description formatted using headings and lists:

1. Network Entity

Relationships

Contains one or more iSCSI Nodes

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

2. iSCSI Node

Relationships

Has a one-to-one relationship with Network Entity

Can be either an iSCSI Target Node or an iSCSI Initiator Node (one or none of each)

3. iSCSI Target Node

Relationships

Has a one-to-one relationship with iSCSI Node

Contains one or more Portal Groups

4. iSCSI Initiator Node

Relationships

Has a one-to-one relationship with iSCSI Node

Contains one or more Portal Groups

5. Portal Group

Relationships

Has a one-to-one relationship with iSCSI Target Node or iSCSI Initiator Node

Contains one or more Network Portals

6. Network Portal

Relationships

Belongs to a Portal Group

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Other Diagrams

We have included an additional set of examples based on some of the SVG diagrams found in the Secure Frame (SFrame) document. The HTML file, "index.html", can be found in the "SVG Examples" folder, along with a CSS stylesheet and some JavaScript utilities. These examples aim to illustrate various methods we've highlighted so far, with the focus on diagrams rendered as inline <svg> elements, but they may not always relate to the previously mentioned diagram types.

SVG with Inline List as Long Description in Disclosure

URL: https://www.ietf.org/archive/id/draft-ietf-sframe-enc-09.html#figure-5

Description:

- The diagram has a short description of: "Diagram illustrating the Decryption Steps listed in the following section.", which points to the accompanying list. See: Short Description: SVGs.
- For illustration purposes, the long description is placed right after the diagram using a disclosure component marked up using <details> and <summary> elements.
- A heading precedes the list of the decryption steps. The heading tag is a direct child of the <summary>.
- The heading is marked up as a <h3> in this case, as a subheading of the "SVG with Inline List as Long Description in Disclosure" level 2 heading inside the sample document. The heading level must be carefully reviewed for each case to ensure that it fits within the overall page structure.
- Steps are mentioned in an ordered list (<ol>) after the diagram. If this is not felt to be an accurate representation, an unordered list may be used instead (<ul>).
- Each step is encompassed in an <li> element.
- Each step is brief and uses plain language as much as possible.

Byte Layout with Inline Table as Long Description

URL: https://www.ietf.org/archive/id/draft-ietf-sframe-enc-09.html#figure-2

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

The diagram illustrates the structure of the SFrame header with a detailed breakdown of the config byte component.

Description:

- The diagram has a short description of: "SFrame Header Structure: The header is divided into three fields. The first field is the "Config Byte", with bit values mentioned in the previous data table. The second field is labeled "KID...", and the third is labeled "CTR...".
- The short description points to the accompanying data table. See: Short Description: SVGs.
- The table is placed before the diagram.
- The section of the table with the config byte structure has the bit number implemented as a row header. The current layout was chosen for the table based on the structure of the SFrame Header described in the section.

Byte Layout with Headings and Table in Dialog as Long Description

URL: https://www.ietf.org/archive/id/draft-ietf-sframe-enc-09.html#figure-2

The diagram illustrates the four different forms of the SFrame header.

Description:

- The diagram has a short description of: "Four forms of Encoded SFrame Header; consult the Figure 3 Diagram Description to review the conditional header structure".
- The short description points to the accompanying dialog with the alternate structured visual description. See: Short Description: SVGs.
- The dialog can be accessed using a button positioned right after the diagram, labelled "Figure 3: Diagram Description". The label may be changed, but it is important to note that a document may have several such buttons, and thus it is important that each button has a unique, informative label which indicates its action and the figure it refers to.
- The overlay is a simple modal dialog using the native <dialog> element.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

    - it starts with a "Close" button;
    - it has a visible title as <h1>; and
    - it is named via aria-labelledby.
- The condition for each header is implemented as <h2>.
- The table structure is similar to the one in the previous example.

SVG with Built-In Semantic Markup and Visually-Hidden Text

URL: https://www.ietf.org/archive/id/draft-ietf-sframe-enc-09.html#figure-1

Section 4.1 describes the encryption process, and a diagram is provided showing actor 1 (Alice) sending a message that goes through Encrypt, Packetize, and Protect processes to reach the media server, then unprotect, depacketize, and decode to reach actor 2 (Bob). In our example we modified the SVG markup to group the steps in a list of seven items.

Description:

- The SVG has role="region" applied and an aria-label attribute to give it an accessible name.
- Most shapes and graphical content are hidden using aria-hidden="true".
- The steps are encompassed in a <g> element with role="list".
- Each step is wrapped in <g> with role="listitem".
- The Key Management text is hidden with aria-hidden="true".
- Text with the "visually-hidden" class is added to describe the Key Management process.
- For each list item with multiple text elements, we are using <tspan> tags to combine individual lines of text into one <text> element. This results in clearer screen reader speech output when navigating sequentially through the list.

A more efficient path is to provide a structured long description with the information illustrated in the diagram in text that is accessible to all users. For example, this can be achieved using a list of seven items to accompany the diagram. See: Unified Modeling Language (UML) Diagrams.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Appendix

Dialog

Modal dialogs must convey their boundaries in a programmatic manner and must allow for elements to be focused in a logical sequence.

To implement an accessible modal dialog, wrap the overlay container in a native <dialog> element which comes with several built-in accessibility features:

- provides programmatic boundaries for the overlay;
- sends focus to the dialog's first focusable element when displayed;
- prevents users from reaching the dimmed main page content when in view; and
- returns the focus back to the triggering control when dismissed (assuming that control still exists within the underlying document).

Additionally:

- Ensure that two separate buttons exist:
  - one within the parent document to trigger the dialog display; and
  - one to dismiss it, placed inside the modal.
- Provide a visible dialog title to describe its content, implemented as a level 1 heading. This further emphasizes the modal nature of the dialog for screen reader users and indicates that the rest of the page content is unavailable.
- Use an aria-labelledby attribute on the <dialog> to provide an accessible name for the overlay. Its value should be set to the ID of the title heading.

Refer to: MDN Web Docs: <dialog>: The Dialog element.

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

Disclosure

A disclosure control toggles the display of an associated piece of content, when activated, and can be used to toggle the display of a long description. It differs from a modal dialog in two key ways:

- there is only one control to toggle the display, rather than one button to expand it and another to collapse; and
- the content area is not modal when expanded, i.e. users can still access the rest of the document.

The easiest way to implement an accessible disclosure component is to use the native <details> and <summary> HTML elements. They come with several built-in features:

- These elements add semantic meaning to your HTML, clearly indicating the purpose of the content. This improves the overall structure of the document and helps search engines understand the content better.
- The <summary> element serves as a toggleable button, which users can expand or collapse to reveal or hide additional content. It is fully operable using the keyboard and other access technologies.
- The expanded and collapsed states of the <details> element are managed by the browser, simplifying the development process. There is no need to write extra JavaScript to handle these states. Screen reader users are accurately informed of the disclosure state.
- Heading markup may be added for the disclosure control by nesting the heading tag inside the <summary> element.
- If desired, a standard Control+F operation on a web page can search inside the collapsed <details> elements on a page and automatically expand one if a match is found.

Using <details> and <summary> is straightforward and requires minimal coding compared to creating custom disclosure components with JavaScript and CSS. Additionally, these elements are well-supported across modern browsers, reducing the need for complex cross-browser compatibility fixes.

Refer to:

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 **www.pac.bz**
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4

- MDN Web Docs: <details>: The Details disclosure element
- MDN Web Docs: <summary>: The Disclosure Summary element

312 Montibello Drive, Cary, NC, USA 27513 919.345.3832 www.pac.bz
1409 Morley Blvd, Ottawa, ON, Canada K2C 1R4