

The QAML FAQ

The QAML F.A.Q.



This FAQ is about QAML, a markup language for internet FAQs.

QAML is the Question and Answer Markup Language.

[Table of Contents]

A. General

[A.1. What is a FAQ?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

A FAQ is the nickname for a "Frequently Asked Questions" document. There are thousands of FAQs available on the World Wide Web on many different topics. The basic format of a FAQ is that someone asks a question, and someone else answers.

FAQs were very popular on specialist bulletin boards, mailing lists and USENET newsgroups. On these discussion

Table of Contents

The QAML F.A.Q.

A. General

- A.1. What is a FAQ?
- A.2. What is a Markup Language?
- A.3. What is QAML?
- A.4. Who wrote QAML?
- A.5. Why not just use HTML?

B. About Writing QAML Documents

- B.1. How can I write a QAML document?
- B.2. What is the smallest QAML document?
- B.3. How can I make a list in QAML?
- B.4. What stylesheets are available for QAML?
- B.5. How can I make a hypertext link?
- B.6. What is the relationship between QAML metadata and Dublin Core?
- B.7. What is the gist attribute?
- B.8. Why is the email address of the author mandatory?
- B.9. How can I include graphics?

C. Processing QAML documents

- C.1. What can I use QAML documents for?
- C.2. What are text processing languages?

D. QAML and Standards

- D.1. Are Standards Useful?
- D.2. What is XML?
- D.3. What is I18n?
- D.4. What is Accessibility?
- D.5. What is Dublin Core?

Cataloging Information (Dublin Core)

forums, the same questions appear repeatedly: newbies ask the same questions and new approaches to solving problems get developed. These questions and answers are collected into FAQs; newbies can be directed to the FAQs for good answers, and the FAQs can get updated.

In several cases, FAQs are ultimately revised into book form and published in paper: *The SGML FAQBook* by Steve DeRose and *Chinese Japanese Korean Vietnamese Information Processing* by Ken Lunde are both examples of FAQs in book form.

You can find a good collection of FAQs at the web site of [faq.org](http://www.faq.org) (<http://www.faq.org/>).

[Table of Contents]

[A.2. What is a Markup Language?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

A markup language lets you name the various elements in your text: HTML is the most famous markup language. You just insert special codes called *tags* into your text; these tags tell you where each interesting element starts or finishes.

The term markup was coined by Dr Charles Goldfarb at IBM in the early 1970s. Goldfarb invented SGML, the Standard Generalized Markup Language (International Standard 8879:1986) which is now becoming available over the WWW as XML, the Extensible Markup Language.

[Table of Contents]

[A.3. What is QAML?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

QAML is a markup language created specially to help write, use and maintain FAQs. It is smaller and simpler than HTML.

A QAML document has a very simple structure: it has a head and a body. The body is made up of `qna` elements, each of which has a `q` question and a answer.

Because the questions and answers are clearly marked up by tags, it makes life a lot easier for people to write programs to manipulate the FAQ: to automatically generate indexes of questions, to make global changes to stylesheets, or to split the questions and answers into different pages.

[Table of Contents]

[A.4. Who wrote QAML?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

QAML version 1.0 was written by Justin Higgins, of faq.org. Email jhiggins@dn.net. This version used SGML, and so is very lenient on tagging errors.

QAML version 2.0 uses XML, and so you need to be more careful with providing all the tags needed for a well-formed XML document. It has very full support for the latest World Wide Web Consortium specifications (see <http://www.w3.org>). It was written by Rick Jelliffe, of the Academia Sinica Computing Centre, Taipei, Taiwan.

[Table of Contents]

[A.5. Why not just use HTML?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

HTML is much more complicated and richer than QAML. However, HTML allows you to create documents which look pretty but which do not have the tagging needed to create indexes or to convert the data to different forms: it gives you too much!

FAQ documents can be accessed from all sorts of systems, not only visual web browsers. So QAML does not provide the types of elements which cannot be seen on many different media: telnet terminals, Braille readers, speech synthesizers, mobile phones, etc.

However, it is very easy to convert QAML to HTML. There are XSL stylesheets available to do this. So you can write your FAQ in QAML, but deliver it in HTML.

[Table of Contents]

B. About Writing QAML Documents

[B.1. How can I write a QAML document?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

You can use any text editor, or SGML or HTML editor. The NotePad or WordPad applications allow it. You can use most word processors (but remember to save the document as "Text". You can even use a WYSIWYG HTML editor to create the FAQ, and then use a text editor to alter the element type names and attributes as appropriate.

There is also an OmniMark script available which does automates much of the HTML-to-QAML conversion work.

A QAML document has a very simple structure: it has a head and a body.

The head has to have some *metadata* which gives the title, the maintainers name and email contact address, the official archive site for the FAQ, and keywords which can be given to a search robot.

The body is made up of `qna` elements, each of which has a `q` question and a answer. These `qna` elements can be collected into sections, and given general section titles.

Within each answer, you can have `p` paragraph elements. The only element types allowed inside a paragraph are `link` and `span`.

- A link acts like a hypertext link in HTML: if you are using a hypertext browser, you should be able to click on it, and go to the location specified. If you are not using a hypertext browser, the link may only appear in a different color. (E.g., at the moment the CSS stylesheets for QAML do not support hypertext.)
- The span element acts like its HTML inspiration. It lets you mark up a range of text inside a paragraph. You give it a class attribute to signify what its meaning is. Typically this will be used to select some font variation, such a bolding. However, please remember to write your FAQ so that it still makes sense to someone who is not using a big color screen and a fancy web browser: use the formatting to *assist* the meaning of the text not to *convey* it.

[Table of Contents]

[B.2 What is the smallest QAML document?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

The following empty document is the smallest QAML 2 document that is valid. You must have at least the FAQ title and one question and answer.

You can use this example as a template for creating new FAQs.

```
<?xml version="1.0" encoding="utf-8" ?>

<!DOCTYPE faq
  PUBLIC "-//IDN faq.org//DTD Frequently Asked Questions 2.4//EN//XML"
  "http://www.ascc.net/xml/resource/qaml-xml.dtd" >

<faq>
  <head><title></title>
</head>
<body>
  <qna id='q1'><q></q>
  <a><p></p>
</a>
</qna>
</body>
</faq>
```

[Table of Contents]

[B.3. How can I make a list in QAML?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

You can use the `<div class="ul">` tag to start a list, and the `</div>` tag to end it. Inside the list, each item can be tagged with a special version of the paragraph tag `<p class="li">`

So here is the markup:

```
<div class="ul">
```

```
<p class="li">AAAA</p>
<p class="li">BBBB</p>
</div>
```

And here is how your browser and your current stylesheet displays it:

- AAAA
- BBBB

QAML is geared towards text systems which may not have sophisticated visual formatting available. This is why there are no tables in QAML. So if you use lists, keep them simple, and make sure the text explains that you are using a lists: some formatters may just show a paragraph as a paragraph.

[Table of Contents]

[B.4. What stylesheets are available for QAML?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

There are CSS and XSL stylesheets available from Academia Sinica Computing Centre's Chinese XML Now! project, at <http://www.ascc.net/xml/en/utf-8/resource-index.html>

[Table of Contents]

[B.5. How can I make a hypertext link?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

The link element has an attribute called href. Just put in the URL of the resource that is the link's target.

Remember that the data inside the link should also contain the URL. This is because some people may read the FAQ from a paper printout. Don't forget them!

An example follows.

```
<p>This is <link href="qaml-faq.xml">qaml-faq.xml</link></p>
```

(Some WWW browsers may display that example wrong, so another version of the same thing follows: the one with the angle brackets is the correct one.)

```
<p>This is <link href="qaml-faq.xml">qaml-faq.xml</link></p>
```

[Table of Contents]

[B.6. What is the relationship between QAML metadata and Dublin Core?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

Dublin Core metadata is a list of some basic metadata categories that is useful for any WWW document authored by a general person. It is "Metadata for the rest of us".

QAML 2 was written with the Dublin Core categories in mind. Because QAML 1 was written before Dublin Core, the same names are not used, but the QAML 2 DTD includes the mappings to Dublin Core elements as comments.

The mappings between Dublin Core elements (with the DC prefix) and QAML 2 element types is given below.

- QAML's title maps to DC.TITLE .

- QAML's maintain maps to DC.CREATOR.
- QAML's author and contributor map to DC.CONTRIBUTOR .
- QAML's archive maps to DC.IDENTIFIER .
- QAML's hdr and althdr map to DC.DESCRPTION .
- QAML's topic and topic maps to DC.SUBJECT .

Other Dublin Core metadata may be available: for example the maintainer may in some cases be regarded as the publisher. Information about the copyrights and extents of the data are not available through standard mappings. Note however that the QAML 2 DTD does include a minimal copyright notice for FAQs: its primary purpose is to remind authors to make copyright policy explicit, and to protect them to a certain extent if they forget.

[Table of Contents]

[B.7. What is the gist attribute?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

The answer element of a qna can have an attribute called `gist`. This attribute should have a one to five word summary of the answer.

This summary may be useful on display devices with limited bandwidth and to aid the user of a QAML document figure out if the answer looks interesting or useful to read further.

[Table of Contents]

[B.8. Why is the email address of the author mandatory?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

The email element is used in fields for authors, contrrrbutors, maintainers. As of QAML 2.4, it is only mandatory for the maintainers.

[Table of Contents]

[B.9. How can I include graphics?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

QAML was simply not designed for rich graphics. Authors are restricted to using fairly simple text. However, two kinds of graphics are available: inline linked graphics and logos.

You can put in inline graphics using the `link` element inside paragraphs. You must also specify an `href` attribute with the WWW URL of the graphic and a `role` attribute with the value "Image". (Another role you may find useful is "Sound".)

You can put in logos using the `logo` element. You can put a logo in as the first element of a `qna`, a section, an author, a contributor and the maintain-er.

These allow logos belonging to the particular person or organization, or to the individual topic. Graphics and logos that belong to a whole series of FAQs are best put into the stylesheet or the FAQ-transformed-into-HTML.

The `logo` element is not intended to be used to include rich graphics. However, if you need it, it is there. But beware that some stylesheets may not print the logo or any links, and some stylesheets may expect the logo to be banner-sized or thumbnail sized.

[Table of Contents]

C. Processing QAML documents

[C.1. What can I use QAML documents for?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

FAQs started on Internet mailing lists and newsgroups. Newcomers ("newbies") often ask the same question. So a FAQ gets newbies up to speed, and stops them from wasting too much time for the grumpy old-timers.

The FAQ format can be a quite useful format for spreading information. Many people find it quite useful because the questions often are real questions that people ask rather than the questions that gurus think should be asked. Also, the answers are fairly targeted to answering particular needs; so FAQs are in a way fairly similar to the "for Dummies" books.

In the case of the QAML FAQ, some of the questions are real, some are faked to

QAML is also being used for inhouse corporate information systems; for companies where user-support is a key component but where the user-liason staff are not specialists, QAML can provide a way to make sure that when someone finds an answer, the answer is available to other people.

[Table of Contents]

[C.2. What are text processing languages?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

A text processing language reads in a text file, recognizes certain delimiters and symbolic tokens, and performs various kinds of transformations based on those tokens.

UNIX has many small text processing languages; these have been bundled together into Perl, which is the most famous general text processing language.

Omnimark is another excellent text processing language, geared towards marked-up documents.

James Clark, of <http://www.jclark.com>, categorizes text processing languages into "push" languages (the events discovered in a document drive trigger function calls) and "pull" languages (the document is read into a data structure like a tree, and then the main function inserts the appropriate parts of the data into a template).

QAML (and other XML and SGML) documents can be processed using general-purpose text processing tools. But they can be also processed using structure-aware tools, such as XP or LotusXSL, which are freely-available implementations of the XSLT language (Extensible Stylesheet Transformation Language).

The most common thing to do with a text processing language and a QAML (or XML) document, is to transform the document in HTML. The HTML can fit into the corporate style or taste of the publisher.

[Table of Contents]

D. QAML and Standards

[D.1. Are Standards Useful?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

Standards are very useful, because they allow us to build amazing things co-operatively.

But a standard is usually general purpose, and a general-purpose tool is usually sub-optimal than a custom-made, specific-purpose tool. So QAML is better for representing FAQs than HTML, but your own markup language will

be even better for marking up **your** own FAQ.

But if you make your own markup language, you have to also then write all your own computer tools to use it. However, QAML is very simple, so even if you do have your own markup language for FAQs, you may find it is not difficult to export your information into QAML format, so that you can use any QAML tools. Or you may find you can convert QAML tools for your requirements too!

[Table of Contents]

[D.2. What is XML?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

XML is the Extensible Markup Language. It is a simple syntax for constructing markup languages to exchange data over the WWW. QAML 2.0 documents are XML documents. (And XML documents are SGML documents.)

See <http://www.w3.org/> for more details.

[Table of Contents]

[D.3. What is I18n?]

Rick Jelliffe,

ricko@gate.sinica.edu.tw

I18n is the contraction used to mean "internationalization", because there are 18 letters missing.

I18n means the features available to allow documents in different languages, perhaps even within the same document.

QAML 2.0 is highly internationalized, following WWW best practise. (One of

QAML 2.0's authors is a member of the World Wide Web Consortium's I18n Interest Group, and proposed most of the i18n features of XML.) It currently does not have support for bi-directional text; this can be added on request.

QAML 2.0 documents can use any standard character encoding, and all elements can have an attribute `xml:lang` to give the language being used.

[\[Table of Contents\]](#)

[D.4. What is Accessibility?]

[Rick Jelliffe](#),

ricko@gate.sinica.edu.tw

Accessibility means that the document can be read by people on unusual mediums or formats: large magnification screens, speech synthesizers, Braille readers and so on.

QAML has been written following the World Wide Web Consortium guidelines in this area: elements with vision-dependencies have kept out, and logo images must have alternative forms.

[\[Table of Contents\]](#)

[D.5. What is Dublin Core?]

[Rick Jelliffe](#),

ricko@gate.sinica.edu.tw

The Dublin Core is a basic set of 13 elements which various experts have decided are a minimum useful list for capturing basic metadata about small documents on the Web. This metadata allows searching by users and rudimentary classification by librarians.

The QAML DTD (Document Type Definition) shows the mappings to the

various Dublin Core elements. Also, we encourage the use of a final section, with class `dc`, with explicit markup of the Dublin Core elements. This assists users of printed versions of your FAQ.

For more information on Dublin Core, there is a small QAML FAQ on Metadata at <http://www.ascc.net/xml/en/utf-8/dc-fg.xml>.

[Table of Contents]

Cataloging Information (Dublin Core)

```

<DC:TITLE      xml:lang="en">The QAML FAQ </DC:TITLE>
<DC:CREATOR    >Rick Jelliffe </DC:CREATOR>
<DC:SUBJECT    xml:lang="en">FAQ, QAML, Frequently Asked Question,
    Question and Answer Markup Language, XML, SGML
    </DC:SUBJECT>
<DC:DESCRIPTION xml:lang="en">Frequently Asked Questions about using QAML,
    the Question and Answer Markup Language </DC:DESCRIPTION>
<DC:PUBLISHER  xml:lang="en">Computing Centre, Academia Sinica, Taiwan </DC:PUBLISHER>
<DC:TYPE       xml:lang="en">Text.Article </DC:TYPE>
<DC:DATE       >1999-04-06 </DC:DATE>
<DC:RIGHTS    >http://www.ascc.net/xml/en/utf-8/legal.html</DC:RIGHTS>

```

[Table of Contents]

This version: ().

Maintained by: [Rick Jelliffe](#).

Archived at: <http://xml.ascc.net/xml/en/utf-8/camel-faq.xml>.

Permanent links: <http://xml.ascc.net/xml/en/utf-8/camel-faq.xml>.