How To Create A Status Section

For Your QT Specification

To: Editors of documents being developed as part of the QT specifications
From: Jim Melton, co-Chair of the XML Query WG (and one of those editors)
Date: 2008-03-20

Most of you (editors) know that all development of QT specs, except for the Errata documents for the 1.0/2.0 specs, has been moved from the …/xsl-query-specs/ tree in CVS to the …/qtspecs/ tree, also in CVS. Some of you have encountered the new *status section generator*, and most of you who have seen it are undoubtedly taken aback. "What happened," I can hear you asking, "to the good old days when somebody (most recently Jim) just wrote the status sections and we pasted them in?"

Those days are gone now. There are several reasons, but the one that you should care most about is the simple fact that there were always errors in text, in markup, or in both, the style of the status sections often varied considerably between documents and document stages, etc. The new status section generator resolves those problems.

But, value comes with cost. And the cost is that somebody has to set a number of parameters used by the status section generator. That can be confusing, I know, so I've provided this documentation to help overcome the confusion and pain.

What is the "status section generator"?

The status section generator is made of two files and a number of parameters. The two files are:
 http://www.w3.org/XML/Group/qtspecs/etc/status-entities.dtd
 http://www.w3.org/XML/Group/qtspecs/etc/status-general.xml

The complete contents of status-general.xml at the time of writing are (with all comments and empty lines removed for readability in this documentation):

```
<status id="&status-section-id;">
  &initial-boilerplate;
  &documents-and-relationships;
  &document-stage;
  &features-at-risk-para;
  &customized-paragraph;
  &implementation-report;
  &changes-para;
  &comments-para;
  &doc-stability;
  &patent-policy-paragraph;
</status>
```

There is no reason for any editor to make changes to that file. (The exception, of course, is when the W3C or the WGs determine that a different structure of the status section is necessary.) It serves as the *status section generator* for all documents, at all stages of development.

How is the status section generator controlled?

I now cover each of those entity references in some detail. You will see that some of those entities are declared directly in the "root" file[1] of your documents, while others are declared in status-entities.dtd and may be controlled by parameters in that "root" file. I use the symbol "➔" to highlight an action that you must take to properly parameterize the status section.

```
    <status id="&status-section-id;">
```

That tag, obviously, starts the status section of a document. The id attribute is set to be the value of an entity, status-section-id, whose value is determined by an entity contained in the internal DTD subset that your "root" file contains. For example:

```
<!ENTITY status-section-id "status">
```

➔ It is your responsibility to include such an entity declaration.

```
      &initial-boilerplate;
```

As you know, every status section must begin with specific boilerplate prescribed by the W3C's PubRules. This entity reference incorporates that boilerplate text. Your source file need not take further notice of that requirement or of the entity being referenced.

```
      &documents-and-relationships;
```

A number of the documents that QT publishes are part of a set that moves in lock step. (These are, of course, Data Model, Functions and Operators, Formal Semantics, Serialization, XPath, XQuery, XQueryX, and XSLT.) Other documents are not part of such a set; examples include Full Text, the Update Facility, and the Scripting Extension, as well as the various Requirements and Use Cases documents.

➔You must declare this entity in your "root" file's internal DTD subset to be one of the following:

```
&not-set-of-documents;
&set-of-documents-11-preREC;
&set-of-documents-11-REC;
```

The first (`&not-set-of-documents;`) is the proper choice for those documents that are not part of a set. The second (`&set-of-documents-11-preREC;`) is appropriate for documents in that set during development, while the third (`&set-of-documents-11-REC;`) is what you'll select when the documents in that set achieve Recommendation status.

```
      &document-stage;
```

All W3C documents go through a series of stages. They are typically (for Recommendation-track documents): First Public Working Draft, ordinary Working Draft (zero or more), Last Call Working Draft (one or more), Candidate Recommendation, Proposed Recommendation, and Recommendation. The status section requirements vary among those stages.

[1] I use the term "root" file to indicate either the single XML source file containing the entire content of a document (for example, xquery-update-use-cases.xml) or the XML file that includes the other XML files containing components of the document (for example, xquery-update.xml).

➔ You must declare this entity in your “root” file’s internal DTD subset to be one of the following:

```
&doc-stage-FPWD;
&doc-stage-WD;
&doc-stage-LC;
&doc-stage-CR;
&doc-stage-PR;
&doc-stage-REC;
```

Those names are, I believe, self-explanatory. In the near future, the status section generator will be updated to incorporate:

```
&doc-stage-WGNOTE;
```

However, in order for those entities to produce appropriate text, a number of other entities must be declared in your “root” file, some only for certain stages. Let’s look at them one at a time:

```
&spec.devby;
```

➔ You must declare this entity in your “root” file. The choices available to you are;

```
&devby.xquery;
&devby.xsl;
&devby.joint1;
&devby.joint2;
```

The first of those is appropriate for documents developed solely by the XML Query Working Group (*e.g.*, XQueryX and the Update Facility), the second for documents developed solely by the XSL WG (*e.g.*, XSLT), the third for most documents developed by both groups (e.g., Data Model), and the fourth for a few documents developed by both groups (*e.g.*, XPath and Serialization). For the XQuery/XPath source file, it is probably best to use a fifth choice: `&xquery-devby;`, which incorporates the `<phrase role=…` markup used in that document.

```
&advancement.statement;
```

➔ You must declare this entity in your “root” file. The choices available to you are;

```
advance.1WG
advance.2WGs
advance.1WG.to-Note
advance.2WGs.to-Note
```

The first is appropriate for REC-track documents developed by only one WG, the second for REC-track documents developed jointly, and the last two are analogous but are intended for non-REC-track documents (such as Requirements and Use Cases).

```
&doc.LC-comments-due;
&doc.CR-comments-due;
&doc.PR-comments-due;
```

➔ If your document is being published as a Last Call WD, then you must set (in your “root” file’s internal DTD subset, of course) the entity `&doc.LC-comments-due;` to the date by which comments must be submitted in order for the WG(s) to guarantee a forma response. If your document is being published as a Candidate Recommendation, then you must set `&doc.CR-comments-due;`, and if it is being published as a Proposed Recommendation, then you must set `&doc.PR-comments-due;`. In all cases, the value of the entity being declared must be of the form “dd MMM yyyy” (*e.g.*, 08 August 2008).

```
&PR-entrance-criteria;
```

➔ If your document is being published as a Candidate Recommendation, then you must set this entity to fully marked-up text (*e.g.*, a paragraph, multiple paragraphs, or paragraphs and lists), that define the criteria that will be used to determine when a transition from CR to PR is appropriate. (There is an example later in this documentation.)

```
&doc.PR-expected;
```

➔ If your document is being published as a Candidate Recommendation, then you must set this entity to the date (in “dd-MMM-yyyy” form) by which the WG expects to have received enough implementation experience to justify a transition to Proposed Recommendation. This date will almost always lag the date by which CR comments are due by weeks to months.

```
        &features-at-risk-para;
```

➔ If your document is being published as a Candidate Recommendation, then you must set this entity to one of the following three choices:

```
&no-features-at-risk;
&one-feature-at-risk;
&multiple-features-at-risk;
```

If there are no features at risk, then the first choice will be your choice. If there is only a single feature at risk, you’ll select the second alternative. Naturally, you’ll choose the last alternative if there are two or more features at risk.

➔ If you choose either the second or third alternatives, you must declare another entity:

```
&features-at-risk;
```

For `&one-feature-at-risk;`, that entity must be set to a word or phrase (marked up if necessary) identifying the single features at risk. For `&multiple-features-at-risk;`, you must declare the value of that entity to be fully marked-up text (perhaps an unordered list: `<ulist> <item>`…`</item>` … `</ulist>`) that identifies the features that the WG(s) determined to be at risk. (There is an example later in this documentation.).

```
        &customized-paragraph;
```

➔ You must declare this entity in your “root” file to be a fully marked-up paragraph (sometimes two paragraphs, rarely more) that differ from all prior versions of the document and all other documents. The text might summarize the changes from the previous WD, or describe the manner in which some notable requirements has been satisfied. Use your imagination. (There is an example later in this documentation.)

```
      &implementation-report;
```

Some of our documents have associated test suites (*e.g.*, XQuery, XSLT, the Update Facility, and Full Text), while others do not (*e.g.*, Data Model and Formal Semantics).

➔ If your document is one that *does not* have an associated test suite, then you must declare that entity to have the value:

```
&no-implementation-report-exists;
```

➔ If, on the other hand, your document is associated with a test suite, then you must declare that entity to be one of the following:

```
&no-implementation-report-or-test-suite-yet;
&no-implementation-report-yet;
&prelim-implementation-report-exists;
&implementation-report-exists;
```

The names are, I believe, self-explanatory.

➔ You must also declare another entity:

```
&test-suite-location;
```

with a value that is the URI at which the test suite (normally, this is the document that introduces the test suite) can be found.

➔If either a preliminary or "real" implementation report exists and you make the appropriate choice above, then you must declare another entity:

```
&implementation-report-location;
```

with a value that is the URI at which the implementation report can be found.

➔ If your document is destined to become a Note, then you must declare that entity to be:

```
&implementation-report-irrelevant;
```

The XQuery/XPath source file, once a test suite is available and implementation reports exist, might best declare that entity to be:

```
&xquery-xpath-impl-para;
```

```
      &changes-para;
```

➔ Depending on the stage of your document, you must declare that entity to have one of the following values:

```
&post.FPWD.changes;
&post.WD.changes;
&post.WD.nochanges;
&post.LC.changes;
&post.LC.nochanges;
&post.CR.changes;
&post.CR.nochanges;
&post.PR.changes;
&post.PR.nochanges;
```

If you select one of the “nochanges” options, you’re done with this entity.

➔ However, if you selected one of the “changes” options, you must also set the value of:

```
&changelog-id;
```

to the value of the id attribute of the change log of your document.

```
      &comments-para;
```

That entity generates the paragraph that tells reviewers where to send comments.

➔ There’s only one thing that you have to do in your “root” file to support generation of this paragraph, and that is to set the value of:

```
&Bugzilla-key;
```

to the short string that we want our reviewers to put in square brackets at the beginning of the subject line of every comment (*e.g.*, XQuery, XQUpdate, FT, *etc*.).

```
      &doc-stability;
```

With this entity, we’re almost done. It controls the generation of the paragraph that sets readers’ expectations about the stability of the document.

You must set the value of this entity to one of the following:

```
&doc-stability-WD;
&doc-stability-LC;
&doc-stability-CR;
&doc-stability-PR;
&doc-stability-REC;
```

Different wording is required for WG Notes, so the near future will see the creation of one more choice:

```
&doc-stability-Note;
```

The final entity contained in status-entities.dtd is:

```
      &patent-policy-paragraph;
```

➔ In order to support generation of the Patent Policy paragraph, you must declare this entity to be one of the following values, depending on how many WGs participated in the development of your document:

```
&ppp-one;
&ppp-two;
```

➔ If you selected `&ppp-one;`, then you must also set the value of one more entity:

```
&disclosure.one;
```

to one of the following choices:

```
&disclosure.xquery;
&disclosure.xsl;
```

The XQuery/XPath source file might better use a value of:

```
&xquery.ppp;
```

instead of `&ppp-one;` or `&ppp-two;`.

How about some examples?

OK, let’s look at the relevant entity declarations for the Candidate Recommendation of the XQuery Update Facility. I recommend that these lines immediately precede the “`]>`” that terminates the internal subset DTD of your “root” file.

I will comment on parts of these entity declarations by inserting XML-style comments at the appropriate places.

```
<!-- The following two lines bring the status section generator -->
<!-- into the document for parameterization and generation      -->
<!ENTITY % status-entities SYSTEM "../../../etc/status-entities.dtd">
%status-entities;

<!-- I prefer to keep all of these dates hanging around, even   -->
<!-- after they are no longer used, for historical purposes     -->
<!-- Note that undetermined future dates have dummy text values -->
<!ENTITY doc.WD-pubdate "07 November 2006">
<!ENTITY doc.LC-pubdate "28 August 2007">
<!ENTITY doc.LC-comments-due "31 October 2007">
<!ENTITY doc.CR-pubdate "14 March 2008">
<!ENTITY doc.CR-comments-due "20 June 2008">
<!ENTITY doc.PR-expected "31 July 2008">
<!ENTITY doc.PR-pubdate "TO BE SPECIFIED">
<!ENTITY doc.PR-comments-due "TO BE SPECIFIED">
<!ENTITY doc.REC-pubdate "TO BE SPECIFIED">
```

```
<!-- The next two entities were not discussed above, because    -->
<!-- they are not directly used by the entities declared in     -->
<!-- entities-general.xml, but they must be set to one of the   -->
<!-- choices above.                                             -->
<!ENTITY doc.pubdate "&doc.CR-pubdate;">
<!ENTITY doc.comments-due "$doc.CR-comments-due;">

<!-- Specify the value of the id attribute to be used           -->
<!ENTITY status-section-id "status">

<!-- Indicate that this document is not part of a set           -->
<!ENTITY documents-and-relationships "&not-set-of-documents;">

<!-- The Update Facility is a Candidate Rec developed solely by -->
<!-- the XML Query WG, and it is on the Recommendation track    -->
<!ENTITY document-stage "&doc-stage-CR;">
<!ENTITY spec-devby "&devby.xquery;">
<!ENTITY advancement.statement "&advance.1WG;">

<!-- Update Facility is CR, so we must give the exit criteria   -->
<!-- Note that this is completely marked up.  Also note the use -->
<!-- "&XQWG;" to indicate that this doc was developed solely by -->
<!-- the XML Query WG; the other choice is "&XSLWG".            -->
<!ENTITY PR-entrance-criteria '<p>The &XQWG; intends to submit
this document for consideration as a W3C
<loc href="http://www.w3.org/2004/02/Process-
20040205/tr.html#RecsPR">Proposed Recommendation</loc>
as soon as the following conditions are met:
</p>
<olist>
<item><p>A test suite is available that tests each identified XQuery
Update Facility feature,
both required and optional.</p></item>
<item><p>Minimal Conformance to this specification, as defined in
      <specref ref="id-minimal-conformance"/>, has been demonstrated by
at least
      two distinct implementations, at least one of which uses the
XQuery human-readable
      syntax defined in this specification.</p></item>
<item><p>The Working Group has responded formally to all issues raised
during
      the CR period against this document.</p></item>
</olist>'>

<!-- Because this is for a CR document that has features at     -->
<!-- risk, we supply a marked up paragraph for that purpose.    -->
<!ENTITY features-at-risk-para "&multiple-features-at-risk;">
<!ENTITY features-at-risk '<ulist>
<item><p><loc href="#dt-revalidation-mode">Revalidation modes other than
"skip"</loc></p></item>
```

```
<item><p><loc href="#id-update-facility-static-typing-feature">the
Update Facility Static Typing Feature</loc></p></item>
<item><p><loc href="#id-xqueryx-update-conformance">XQueryX
Conformance</loc></p></item>
</ulist>'>

<!-- Here is our custom paragraph                               -->
<!ENTITY customized-paragraph '<p>The WG
believes that this document, published on &doc.pubdate;,
is sufficiently mature and stable that the development
community can begin developing implementation experience
and reporting on that experience. </p>
<p>The WG particularly solicits comment on a key design principle of
this specification:
As indicated in <specref ref="id-processing-model"/>, expressions may
return <emph>either</emph>
a value <emph>or</emph> a pending update list, but not both.
We specifically solicit feedback on that decision. </p>'>

<!-- CR documents should identify a test suite; it’s really     -->
<!-- if they can also identify a preliminary impl. report.      -->
<!-- Note the use of "&report-public;". The other choice is     -->
<!-- "&report-member-only;".                                    -->
<!ENTITY implementation-report '&prelim-implementation-report-exists;'>
<!ENTITY test-suite-location
    "http://dev.w3.org/2007/xquery-update-10-test-suite/">
<!ENTITY implementation-report-availability "&report-public;">
<!ENTITY implementation-report-location
    "http://dev.w3.org/2007/xquery-update-10-test-suite/results/">

<!-- Because this is a CR document that has had changes since   -->
<!-- end of the Last Call draft, we indicate those changes here -->
<!ENTITY changes-para "&post.LC.changes;">
<!ENTITY changelog-id "id-revision-log">

<!-- So the comments paragraph can tell reviewers how to        -->
<!-- identify the document on which they're commenting...       -->
<!ENTITY Bugzilla-key "UPD">

<!-- And, of course, we must indicate that this document has    -->
<!-- the stability of a CR document.                            -->
<!ENTITY doc-stability "&doc-stability-CR;">

<!-- Finally, we set up generation of the patent policy para    -->
<!ENTITY disclosure.one "&disclosure.xquery;">
<!ENTITY patent-policy-paragraph "&ppp-one;">

<!-- The next line must follow all the previous ones, and I     -->
<!-- recommend that it be the last in your internal DTD subset  -->
<!ENTITY status-section SYSTEM "../../../etc/status-general.xml">
```

Conclusion

It can be difficult to set up your first document’s status section using the status section generator, but it’s pretty easy to maintain as the document progresses. This documentation should ease the difficulties at least somewhat, but you should feel free to ask questions as long as you need in order to understand this. I am also quite willing to set up your status sections for you initially.