

Data-centric XML

Atom

Atom 1.0

- Syndication format (definition of "feeds")
 - mix between "data" and "document"
- RFC 4287
- Namespace: <http://www.w3.org/2005/Atom>
- defined using RelaxNG
- extensible format
- two kinds of documents: feeds and entries
- structure:
 - common data
 - containers
 - metadata
 - digital signature and encryption
 - extensions

Example Feed

```
<feed xmlns="http://www.w3.org/2005/Atom">

    <title>Example Feed</title>
    <link href="http://example.org/" />
    <updated>2003-12-13T18:30:02Z</updated>
    <author>
        <name>John Doe</name>
    </author>
    <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

    <entry>
        <title>Atom-Powered Robots Run Amok</title>
        <link href="http://example.org/2003/12/13/atom03" />
        <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <updated>2003-12-13T18:30:02Z</updated>
        <summary>Some text.</summary>
    </entry>

</feed>
```

Text Constructs

- **human-readable text** (plain text, html, or xhtml)
 - html: **no** child elements - rather, HTML source
 - html: text should be usable as content of a div element
 - xhtml: single xhtml:div element

atomPlainTextConstruct =

```
atomCommonAttributes,  
attribute type { "text" | "html" }?,  
text
```

atomXHTMLTextConstruct =

```
atomCommonAttributes,  
attribute type { "xhtml" },  
xhtmlDiv
```

atomTextConstruct = atomPlainTextConstruct | atomXHTMLTextConstruct

Person Constructs

```
atomPersonConstruct =  
    atomCommonAttributes,  
    (element atom:name { text }  
    & element atom:uri { atomUri }?)?  
    & element atom:email { atomEmailAddress }?)?  
    & extensionElement*)
```

Date Constructs

- xsd:dateTime, also conforming to RFC 3339 (Date and Time on the Internet: Timestamps)
 - implicitly conforming to ISO 8601, W3C note on datetime, and XML Schema

Feeds

```
atomFeed =  
    element atom:feed {  
        atomCommonAttributes,  
        (atomAuthor*  
        & atomCategory*  
        & atomContributor*  
        & atomGenerator?  
        & atomIcon?  
        & atomId  
        & atomLink*  
        & atomLogo?  
        & atomRights?  
        & atomSubtitle?  
        & atomTitle  
        & atomUpdated  
        & extensionElement*),  
        atomEntry*  
    }
```

Feed (2)

- various constraints not expressed in RelaxNG
 - e.g. atom:feed must contain atom:author, unless all atom:entry contain atleast one atom:author
- text recommendations per entry to support proper rendering
 - title should be present and non-empty
 - content should be non-empty if present
 - summary should be present if content is absent

Entry

```
atomEntry =  
    element atom:entry {  
        atomCommonAttributes,  
        (atomAuthor*  
        & atomCategory*  
        & atomContent?  
        & atomContributor*  
        & atomId  
        & atomLink*  
        & atomPublished?  
        & atomRights?  
        & atomSource?  
        & atomSummary?  
        & atomTitle  
        & atomUpdated  
        & extensionElement*)  
    }
```

Entry (2)

- Content: either inline (text content) or out-of-line (src URL + MIME type)

Metadata

```
atomAuthor = element atom:author { atomPersonConstruct }
atomCategory =
  element atom:category {
    atomCommonAttributes,
    attribute term { text },
    attribute scheme { atomUri }?,
    attribute label { text }?,
    undefinedContent
  }
atomGenerator = element atom:generator {
  atomCommonAttributes,
  attribute uri { atomUri }?,
  attribute version { text }?,
  text
}
atomIcon = element atom:icon {
  atomCommonAttributes,
  (atomUri)
}
atomId = element atom:id {
  atomCommonAttributes,
  (atomUri)
}
```

Metadata (2)

```
atomSource =  
element atom:source {  
    atomCommonAttributes,  
    (atomAuthor*  
     & atomCategory*  
     & atomContributor*  
     & atomGenerator?  
     & atomIcon?  
     & atomId?  
     & atomLink*  
     & atomLogo?  
     & atomRights?  
     & atomSubtitle?  
     & atomTitle?  
     & atomUpdated?  
     & extensionElement*)  
}  
...
```

Extensions

- Markup outside of atom namespace allowed at various places
 - "unknown foreign markup" MUST be accepted by Atom processors; SHOULD be ignored inside Text Constructs or atom:content

IANA Considerations

- MIME media type name: application
 - MIME subtype name: atom+xml
 - Mandatory parameters: None.
 - Optional parameters: "charset"
 - ...
 - File extension: .atom
 - Macintosh file type: TEXT
 - Contact person: Mark Nottingham
 - Intended usage: COMMON
 - Author/Change controller: IESG
- IANA maintains registry of link relations (initially: alternate, related, self, enclosure, via)