

RDF Site Summary 1.0 Modules

Authors

The members of the RSS-DEV Working Group:-

[Gabe Beget-Doy, JFinity Systems LLC](#)
[Dan Brickley, ILRT](#)
[Rael Dornfest, O'Reilly & Associates](#)
[Ian Davis, Calaba, Ltd.](#)
[Leigh Dodds, xmlhack](#)
[Jonathan Eisenzopf, Whirlwind Interactive](#)
[David Galbraith, Moreover.com](#)
[R.V. Guha, guha.com](#)
[Ken MacLeod, \(Independent\)](#)
[Eric Miller, Online Computer Library Center, Inc.](#)
[Aaron Swartz, The Info Network](#)
[Eric van der Vlist, Dvomedea](#)

Version

[1.3.2 2001-03-20](#) Added content module ([as](#))
1.3.1 2000-12-20 (editor: [SBP](#) - cleaned up markup)
1.3 2000-12-09

Status

[Release](#)

Rights

Copyright © 2000 by the Authors.

Permission to use, copy, modify and distribute the RDF Site Summary 1.0 Specification and its accompanying documentation for any purpose and without fee is hereby granted in perpetuity, provided that the above copyright notice and this paragraph appear in all copies. The copyright holders make no representation about the suitability of the specification for any purpose. It is provided "as is" without expressed or implied warranty.

This copyright applies to the RDF Site Summary 1.0 Specification and accompanying documentation and does not extend to the RSS format itself.

Abstract

RSS modules are XML-namespace based compartmentalized extensions to [RDF Site Summary \(RSS\) 1.0](#).

Table of Contents

- [1. Introduction](#)
- [2. Modules](#)
- [3. Module Guidelines](#)
 - [3.1 XML Namespaces](#)
 - [3.2 Simple vs. Rich Content Models](#)
 - [3.3 RDF](#)
 - [3.4 Compatibility](#)
 - [3.5 Formats](#)
- [4. Resources](#)

1. Introduction

Namespace-based modularization affords for compartmentalized extensibility, allowing RDF Site Summary (RSS) 1.0 to be extended:

- without need of iterative rewrites of the core specification
- without need of consensus on each and every element
- without bloating RSS with elements the majority of which won't be used in any particular arena or application
- without namespace collisions

The only modules that ship "in the box" with RSS 1.0 are [Dublin Core](#) and [Syndication](#).

This document serves as a registry for modules discussed on and adopted by the members of the [RSS-DEV Mailing List](#). Some examples of module usage may be found in the [Examples](#) of the [RSS 1.0 Specification](#).

2. Modules

RSS 1.0 modules are maintained in separate documents, available online at <http://purl.org/rss/1.0/modules/>. Modules are classified as [Proposed](#) until accepted as [Standard](#) by members of the RSS-DEV working group or a sub-membership thereof focused on the area addressed by the module.

The current set of standard RSS 1.0 modules is:

- [Dublin Core](#)
- [Syndication](#)
- [Content](#)

3. Module Guidelines

3.1 XML Namespaces

Every RSS 1.0 module should reside in its own [XML Namespaces](#) and compartmentalize some grouping of functionality with a specific purpose in mind. A module should be as narrowly defined so as to avoid the "kitchen sink" feel while taking care to be inclusive of all the functionality required to serve the purpose at hand.

3.2 Simple vs. Rich Content Models

Module authors are left to choose whether the task at hand demands a simple or rich content model. Adopting a simple model, however, does not preclude later semantic augmentation -- either to the existing module itself or via a secondary module.

An Example: The Dublin Core module defines a simple flat structure of elements containing string literals:

```
...  
<dc:subject>Woodworking</dc:subject>  
<dc:publisher>John Doe</dc:publisher>
```

```
<dc:creator rdf:about="dc:creator/"
...

```

The subject and creator are simply the string literals "Woodworking" and "Arthur Author," respectively. At some point, however, additional semantics for dc:subject may be overlaid by way of a taxonomy module:

```
...
<dc:subject>
  <taxo:topic rdf:about="http://dmoz.org/Arts/Crafts/Wood_Craft/Woodworking/">
    <rdf:value="Woodworking" />
  </taxo:topic>
</dc:subject>
...

```

Now subject is defined by a resource, here a branch of the [Open Directory Project](#) (DMOZ) tree, rdf:value providing a default string literal value of "Woodworking" for non-RDF parsers or applications needing only a simple bit of text as a subject description--this is termed the "dumb down" principle in some circles.

Application developers are advised to expect such semantic augmentation. Module developers are advised to make it clear how plain content is to be extracted from richer models.

3.3 RDF

While modules are not required to take full advantage of RSS 1.0's RDF framework, effort should at least be made to construct modules that are at least RDF-compliant and do their best to provide coherent data models for both RDF and plain XML.

With this in mind, any element containing XML markup (i.e. other elements) that is not written as RDF should signal RDF parsers that this markup should not be interpreted, instead maintained as a value using the `parseType="Literal"` attribute.

A simple example employing `parseType="Literal"`:

```
<dc:creator rdf:parseType="Literal">
  <name>
    <firstname>John</firstname>
    <middle_initial>Q.</middle_initial>
    <lastname>Public</lastname>
  </name>
</dc:creator>

```

3.4 Compatibility

RSS modules must not introduce conflicts by ad hoc modification of the content models of any other module or the core. Modular extensions may not be considered stand-ins for required core elements (eg. dc:description at the channel level does not obviate the need for including the required rss:description element).

3.5 Formats

Date

RSS modules are to adopt the the W3C's preferred date and time formats [\[W3CDTF\]](#).

4. Resources

Please see the [Resources](#) section of the [RSS 1.0 Specification Proposal](#).