

Registration Procedures for S0IF Template Types

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

The Summary Object Interchange Format [Ref. 1] was first defined by the Harvest Project [Ref 2.] in January 1994. S0IF was derived from a combination of the Internet Anonymous FTP Archives IETF Working Group (IAFA) templates [Ref 3.] and the BibTeX bibliography format [Ref 4.]. The combination was originally noted for its advantages of providing a convenient and intuitive way for delimiting objects within a stream, and setting apart the URL for easy object access or invocation, while still preserving compatibility with IAFA templates.

S0IF uses named template types to indicate the attributes which may be contained within a particular summary object. Within the context of a single application, private agreement on the definition of template types has been adequate. As S0IF objects are moved among applications, however, the need for standard, well-specified, and easily identifiable template types increases. This need is particularly intense in the context of query referral, where knowledge of an attribute's definition and the allowed data types for specific values is crucial. For a discussion of this in the context of the Common Indexing Protocol, see [Ref. 1].

The registration procedure described in this document is specific to S0IF template types. There is ongoing work within the IETF to specify a more generic schema registration facility[Ref. 5]. It is not yet clear whether the results of that work will encompass the ability to register entities like S0IF template types. If it does so, the registration of S0IF template types may be shifted to that method and registry. Should that occur, appropriate pointers will be created in cooperation with the Registrar to ensure that no registrations are lost.

1. Registrar

The initial registrar of SOIF template types will be the Internet Assigned Numbers Authority (IANA).

2. Defining Template Types

Each SOIF object is composed of 3 fundamental components: a template type IDENTIFIER, a URL, and zero or more ATTRIBUTE-VALUE pairs. See [Ref 1.] for the formal grammar of SOIF and a description of how these components interrelate. As part of the registration process, registrants must: propose a template type IDENTIFIER; list the ATTRIBUTES which the template may contain; identify whether each ATTRIBUTE is mandatory or optional; and specify the data type and encoding appropriate for the VALUES associated with each ATTRIBUTE.

2.1 The template type IDENTIFIER

The IDENTIFIER for the template type is assigned at registration based on a proposal from the registrant. It is, however, at the sole discretion of the registrars to assign specific IDENTIFIERS. While they will normally assign the IDENTIFIERS proposed by registrants, they may choose to modify a proposed IDENTIFIER to avoid conflict with other existing or proposed template types.

Because of the pre-installed base of servers using privately agreed upon template types, applications using SOIF need to be able to ascertain whether a referenced template type has been registered. In order to accomplish this, all template type IDENTIFIERS for template types registered with the IANA will begin with the ASCII string "IANA-". An IANA-registered template type based on the GILS specification, for example, might be registered as "IANA-GILS". Should other registrars emerge over time, similar strings must be established and used to compose template type IDENTIFIERS which they assign.

2.2 The URL

The URL associated with a particular summary object is determined by the application generating the object. Applications must generate valid URLs according to the rules of [Ref 6.], but there is no restriction on what sorts of URLs may be associated with particular template types. The use of a particular template type indicates the type of information contained in the summary object, not how the initial resource being summarized was accessed. This aspect of SOIF summary objects is therefore not subject to registration.

2.3 ATTRIBUTES

Where an ATTRIBUTE associated with a proposed template type exactly matches an ATTRIBUTE previously defined in a registered template type, the proposed ATTRIBUTE should be defined by reference to the existing, registered ATTRIBUTE. This allows query referral meshes to easily map queries against ATTRIBUTES derived from different template types and provides an easy method for extending or restricting an existing template type to match an application's particular needs. In such cases, the ATTRIBUTE for the newly registered template type will have the same name, description, and allowed values as the ATTRIBUTE in the existing registered template type.

Where no existing ATTRIBUTE may be referenced, registrants must specify each ATTRIBUTE's name, description, and allowed values.

2.3.1 ATTRIBUTE names

To handle multiple VALUES for the same ATTRIBUTE, SOIF uses a naming convention, appending a hyphen and a positive integer to the base ATTRIBUTE name to create a unique ATTRIBUTE IDENTIFIER. For example, the ATTRIBUTE IDENTIFIERS "Publisher-1", "Publisher-2", and "Publisher-3" can be used to associate three VALUES with the ATTRIBUTE named "Publisher". In order to provide for the unimpeded operation of this convention, ATTRIBUTE names may not terminate with a hyphen followed by an integer. ATTRIBUTE names are otherwise restricted only by the grammar defined in [Ref. 1].

In general, registrants will probably wish to propose ATTRIBUTE names which are short, mnemonic, and intuitively associated with the characteristic that the ATTRIBUTE describes. While these may be generally laudable goals, it must be remembered that the application interface need not present the raw ATTRIBUTE name to the end user; indeed, in situations where the end user's language does not use the ASCII character set, the interface must map the ATTRIBUTE name to an appropriate local representation. Since ATTRIBUTE definitions are provided as part of the registration process, registrants should avoid attempting to overload the ATTRIBUTE name with information which belongs in the description.

2.3.2 ATTRIBUTE descriptions

ATTRIBUTE descriptions for ATTRIBUTES registered with the IANA must be in English, though mappings to other languages may be proposed as part of the ATTRIBUTE description. ATTRIBUTE descriptions should propose clear criteria for establishing whether an object possesses a particular ATTRIBUTE. Descriptions should also include at least two examples of how each attribute relates to an object being summarized,

using, where possible, objects which are broadly available to a wide variety of audiences. If several ATTRIBUTES within a template type inter-relate, the descriptions of each may reference the others; care must be taken, however, that the resulting descriptions are not circular. Where fully realized specifications of the ATTRIBUTES have been created in other contexts, the salient text from those descriptions should be quoted and appropriate references cited.

2.3.3 Required and Optional Attributes

Each ATTRIBUTE registered for a template type must be marked either required or optional. Note that marking an ATTRIBUTE required does not imply that it may not have a null value; it implies only that it must appear in all templates of that registered template type.

2.4 VALUES

For each ATTRIBUTE, the registrant must specify the data format and, if appropriate, the language, character set, and encoding. Where possible, the registrant should include references to a precise and openly available specification of the format. The registrant must also specify the appropriate matching semantics for the ATTRIBUTE if these are not strictly implied by the data format and encoding. The registrant must also note whether null values are permitted.

3. Versioning

Creating a revision of a template type is functionally similar to creating a new template type. A Registrant may propose as a name any derivative allowed under the rules of section 4.1 and [Ref. 1] to the new template type. ATTRIBUTES retained across versions without modification should be referenced as described in section 4.3. Modified ATTRIBUTES must be described as if new. A registrant may note a relationship between a proposed template type and an existing template type as part of the registration process. The following three relationships are currently defined:

Successor: for proposed template types intended to replace an existing template type.

Variant: for proposed template types whose ATTRIBUTES are either a superset or a subset of an existing template type.

Alternate: for proposed template types which share a large number of ATTRIBUTES with an existing template type but whose ATTRIBUTES do not form a strict superset or subset of an existing template type.

Note that there may be relationships between ATTRIBUTES of different template types without there being a named relationship between the template types themselves.

4. Security

S0IF template types which are intended for applications which will pass summary objects over the global Internet should contain authentication ATTRIBUTES. S0IF summary objects lacking authentication ATTRIBUTES must be treated as unreliable indicators of the referenced resource's content and should only be used where other aspects of the environment provide sufficient security to prevent spoofing. Given, however, that particular template types may be intended for environments with such security, there is no requirement that registered template types contain authentication ATTRIBUTES. The application developer must select or propose a template type appropriate for the intended application environment; if none is available with suitable authentication ATTRIBUTES, the provisions of section 4.3 make it easy for the developer to propose an extension to an existing template type with the appropriate authentication ATTRIBUTES.

5. References

- [1] Hardie, T., Bowman, M., Hardy, D., Schwartz, M. and D. Wessels, "CIP Index Object Format for S0IF Objects", RFC 2655, August 1999.
- [2] The Harvest Information Discovery and Access System:
<URL:<http://harvest.transarc.com/>>.
- [3] D. Beckett, IAFA Templates in Use as Internet Metadata, 4th Int'l WWW Conference, December 1995,
<URL:<http://www.hensa.ac.uk/tools/www/iafatools/>>
- [4] L. Lamport, LaTeX: A Document Preparation System, Addison-Wesley, Reading, Mass., 1986.
- [5] IETF Schema Registration Working Group.
<URL:http://www.ietf.org/html.charters/wg.dir#Applications_Area>
- [6] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.

6. Author's Address

**Ted Hardie
Equinix
901 Marshall Street
Redwood City, CA 94063 USA**

EMail: hardie@equinix.com

d) Required ☐ or Optional ☐?

e) Data Type and encoding for this VALUE _____

f) If a specific language and character set are expected, list them here _____

g) Is a null value permitted? Yes ☐ No ☐

7. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.