fhiso

CFPS 37
(Call for Papers Submission number 37)

Proposal for a scalable extensibility mechanism

Submitted by: Smith, Richard

Type: Technical proposal

Created: 2013-04-12

Last updated: 2015-05-30

URL: Most recent version: http://fhiso.org/files/cfp/cfps37.pdf
This version: http://fhiso.org/files/cfp/cfps37_v2-0.pdf

Description: Proposal to address certain extensibility requirements by using XML namespaces, both in XML formats and in non-XML formats such as GEDCOM.

Keywords: extensibility, vendor extensions, forwards compatibility, decentralised development, XML namespaces, URIs, GEDCOM

Change Log for CFPS 37

2015-05-30 cfps37_v2-0.pdf
Added two new sections detailing the proposed application to GEDCOM.

2013-04-12 cfps37_v1-0.pdf
Initial version

Abstract

This paper examines the importance of extensibility in a genealogical standard and indentifies three key principles for an extensibility mechanism. It examines current practice in GEDCOM and how it fails these principles. The extensibility mechanisms used in some other fields are also considered, and in particular XML's. The paper proposes the adoption of XML Namespaces to provide for extensibility in a FHISO standard, notwithstanding the fact that FHISO may adopt a non-XML serialisation format. Such a scheme requires the use of URIs to identify vendor-specific namespaces, and the paper discusses how these might be used.

The FHISO Board have said that one option under serious consideration is that FHISO should define a series of extensions to the current GEDCOM standard. The paper concludes with a look at how the XML Namespace mechanism can be applied to GEDCOM tags while retaining backwards compatibility with the current standard and existing vendor extensions.

1 Introduction

There several reasons why a new genealogical standard should have a well-defined mechansim for extension. These reasons can be divided into three broad groups:

— accommodating vendor-specific extensions;
— anticipating the need for forwards compatibility with a later standard; and
— recognising the modularity required in certain expert systems.

The first case, that of vendor-specific extensions, is the simplest. Experience in any fields, not just genealogy, suggests that vendors will find reasons for wanting to extend a standard. The business need to provide their product with a unique selling point is a frequent reason. Such extensions might be to store additional metadata relevant to that vendors' application. An example can be found in the common `_UID` extension to GEDCOM [1]. This example also highlights a problem with such extensions: they are used incompatibly by different vendors, with different formatting or semantics assumed in different applications. Such naming conflicts can cause problems migrating data between applications.

But good support for extensions is particularly important to a genealogical standard because the size and complexity of the field. No standard can feasibly hope to standardise everything, and therefore vendors are likely to experiment with a new features, perhaps with a view that they may eventually be standardised. The GEDCOM 5.5 EL extensions that improve GEDCOM 5.5's handling of places is an example of that [2]. In the future, FHISO or a successor organisation may decide to standardise certain of these vendor extensions; and FHISO may also wish to develop the standard itself in new ways. Irrespective of how such developments

arise, if a proposed FHISO standard gains traction within the community, it seems certain that the first version will be succeeded by a second version.

Finally, certain areas are inherently open-ended and an extensible, modular approach bests suits them. The world has seen scores, probably hundreds, of calendar systems. It is implausible to suggest that FHISO can standardise each of these, but it is also unreasonable to expect vendors to only ever support the standard ones. An Icelandic vendor may well want to support the old Icelandic calendar, for instance. Personal names have many different possible components, and a standard should accommodate the tagging of matronymics or Japanese postumous names, even though they may not be standardised. Similar consideration apply to geopolitical data, such as the types of subnational divisions.

2 Principles

Avoid name conflicts
: There should be no danger of conflict between extensions made by different vendors. Nor should a vendor extension conflict with future extensions made by FHISO or its successors. This should also apply to the likes of calendars: one vendor's implementation of the old Icelandic calendar may not be compatible with another vendor's.

Easy reuse of extensions
: Where one vendor reuses an extension first introduced by another vendor, it should be able to indicate that this is the case. This should be possible without action or consent from the original implementation. (This is a technical requirement as this paper does not consider legal issues associated with patents or copyright.)

Decentralised development of extensions
: The logistical barrier to creating an extension should be low. An individual developer wishing to develop an extension should be able to do so *in vacuo*, without registering the extension with a central authority such as FHISO.

3 Current practice

Where extensions have been made to GEDCOM outside the official standard, there is an informal convention that the tags associated with extension are prefixed with an underscore. This avoids name conflict between vendor extensions and future standarisation, but as the example of `_UID` demonstrated, it does not prevent conflict between vendors [1].

In certain relevant ISO standards, such as ISO 639 language codes, allocation of

identifiers is overseen by a central registration authority (the Library of Congress for ISO 639), and new codes are only allocated after application to and review by the registration authority [3]. Private extension are permitted with a '`x-`' prefix, but there is no means for preventing conflict between extensions.

In a field as self-contained as the enumeration of languages and with the resources of Library of Congress, it is reasonable to hope that most languages will be identified and tagged, and the need for private extensions therefore rare. Genealogy, as a whole, is not as self-contained, nor are the resources of FHISO comparable to those of the Library of Congress. Consequentially, the need for private extension is therefore much greater as is the risk of conflict between extensions.

Many languages based on XML have solved this problem using XML Namespaces [4]. Element names (and in some circumstances, other names) have an optional prefix, separated from the main part of the name by a colon. Such a name is referred to as a QName, and `x:name` and `name` are example with and without a prefix. The prefix is short-hand for a lengthier unique identifier (a URI, specifically) to which the prefix must be bound in a specified manner.

The XML Namespaces mechanism for using QNames as tags and binding prefixes to URIs has been reused in several formats that are not based on XML. The World Wide Web Consortium's CURIE standard for compact URIs uses it [5], as does the Turtle language [6].

4 General proposal

This paper proposes that a future genealogical standard should mandate that any extensions should be made in a vendor-specific namespace. Each namespace must be formally identified by a URI and references to vendor-specific concepts must be via a prefixed QName, as defined by the XML Namespaces specification [4]. The serialisation format adopted must provide syntax for binding prefixes to namespace URIs, and in the case that an XML serialisation is adopted by FHISO, the namespace binding mechanism in XML Namespaces must be used.

This paper does not seek to influence what the choice of serialisation format should be, and the proposal to adopt one mechanism from XML does not make it necessary to use XML for serialisation. Where extensible tag names are required for the likes of calendars, it is suggested that they too should be QNames. Any standardised calendars can be unprefixed, while non-standard ones can go in a vendor namespace.

For consistency, this paper proposes that FHISO allocate a namespace URI to the *standard namespace* (i.e. the namespace of FHISO-defined terms). Whether any

form of declaration is required to use the standard namespace in a non-XML format is a implementation detail of that serialisation format, and is not considered here.

In XML, two QNames are the same if they have the same local part and the same namespace URI (or none) — that is, the specific choice of prefix conveys no semantic meaning. Although XML Namespaces standard does not provide a single URI represention of a QName, many derived standards (such as CURIE [5] or Turtle [6]) do so by simply concatenating the namespace URI and local part. This paper proposes FHISO does likewise.

5 Namespace URIs

The use of URIs to identify namespaces has several advantages, not least of which is that it is already standardised [7]. Many URI schemes (and particularly the most common `http` scheme) use domain names, and mechanisms already exist for the allocation of domain names. For developers who do not have a suitable domain name, sites such as `http://purl.org/` exist to provide people with persistent URIs. In the event that a developer has good reason not to want to use a domain-based URI for the namespace, a mechanism exists for using UUIDs as URIs [8]. However, this paper proposes that a FHISO standard should recommend (but not mandate) the use of `http` URIs.

No requirement is made by the XML Namespaces standard that the namespace URI can be used to fetch any document or resource. It is simply an identifier. Nevertheless the URI is commonly used as a means of fetching documentation on the namespace, and this paper proposes that FHISO recommends (but not mandate) that, if the vendor wishes to make documentation available on the extensions provided in the namespace, it should be accessible from that URI.

6 Application to GEDCOM

In GEDCOM 5.5, "a *tag* consists of a variable length sequence of *alphanum* characters," where an "alphanum character" is defined as any character from the POSIX character class `[A-Za-z0-9_]`. Tags are explicitly not limited to three or four characters long: "systems should prepare to handle user tags of greater length." This paper proposes that the syntactic space of GEDCOM tag names is partitioned in three as follows.

Standard tags are those defined in the GEDCOM standard, or a future FHISO version of it. This paper proposes that no tags containing an underscore should ever be standardised, which is consistent with past and current practice. They should therefore match the POSIX regular expression `[A-Za-z0-9]+`. `INDI`, `SEX`

and FAMC are all examples of standard tags. Syntactically XYZZY is also a standard tag, even though such a tag has never been defined in a standard: it should not therefore be used unless it is defined in a future FHISO GEDCOM standard.

Unprefixed extension tags are the extensions in use today, and match the POSIX regular expression `_[A-Za-z0-9_]+`. The common _UID extension is an example of one. Some extensions, such as the _PLACE_TYPE tag used in the Personal Ancestral File, include an underscore internally, and this paper continues to allow that [9].

Prefixed extension tags are a new class of tags proposed in this paper and match the POSIX regular expression `[A-Za-z0-9]+_[A-Za-z0-9]+`. As they do not begin with an underscore, they should not conflict with any existing extensions,* and no tags of this form have been standardised to date. They are essentially XML QNames, but written in a GEDCOM-compatible syntax. The part of the tag before the underscore is the *prefix*, and the part after it is the *local part*. An example prefixed extension tag is FS_ID, where FS is the prefix and ID the local part.

The local part of an unprefixed extension tag is defined as the whole tag excluding the underscore, and the local part of a standard tag is defined to be the whole tag.

7 Binding prefixes in GEDCOM

This paper proposes a new standard tag, PRFX, to be placed in the HEAD record and used to declare prefixes and bind them to their namespace URI. The PRFX line value — that is, the content following the PRFX tag — consists of optionally the prefix followed by a space, followed by the namespace URI. If the prefix is omitted, the namespace URI is bound to the *default namespace* as used by unprefixed extension tags. It is not proposed that the URI should be enclosed in angle brackets, as this it does not appear to be required in the WWW tag in the GEDCOM 5.5.1 draft [10]. The following example binds both the default namespace and the GEO prefix to example namespace URIs.

```
0 HEAD
1 GEDC
2 FHISO
1 PRFX http://example.com/gedcom-el/
1 PRFX GEO http://example.com/geospatial/
0 @L1@ _LOC
```

* Some vendors have not stuck to the rule that extensions must begin with an underscore, but a search through lists on the Internet of known extensions finds none with just an internal underscore [9]. Were a serious conflict with existing extensions to be discovered, the single underscore in the prefixed extension tag syntax could be doubled up.

```
1 NAME London
1 GEO_LAT 51.507222
1 GEO_LONG -0.1275
```

The FHISO tag is intended as a way of saying that the file is in a future FHISO dialect of GEDCOM, which for the purpose of this paper adds just the FHISO and PRFX standard tags. It is not a core part of this proposal, and may not be thought necessary.

Per §4, every tag can be mapped to a URI by concatenating the namespace URI with the tag's local part, and this paper recommends that it is this URI that applications use to determine the meaning of the tag, so as to avoid attaching significance to the prefix (or lack thereof). For the sake of exposition, the GEDCOM syntax's standard namespace URI is taken as <http://fhiso.org/gedcom-tags/>.

```
HEAD      <http://fhiso.org/gedcom-tags/HEAD>
FHISO     <http://fhiso.org/gedcom-tags/FHISO>
_LOC      <http://example.com/gedcom-el/LOC>
GEO_LAT   <http://example.com/geospatial/LAT>
GEO_LONG  <http://example.com/geospatial/LONG>
```

This paper suggests that it should illegal to use a prefixed extension tag unless the prefix has first been bound to a *namespace URI*, just as in XML. Because of the need for compatibility with existing extensions, it is likely only possible to deprecate the use of unprefixed extension tags without first binding the default namespace. This paper leaves the mapping of an unprefixed extension tag to a URI undefined while the default namespace is unbound.

References

[1] Tamura Jones, 2012, *The _UID tag — Common GEDCOM Extension* (blog entry), http://www.tamurajones.net/The_UIDTag.xhtml

[2] GenWiki website, 2009, *Gedcom 5.5EL*, http://wiki-en.genealogy.net/Gedcom_5.5EL

[3] Internet Engineering Task Force, 2009, *Tags for Identifying Languages* (RFC 5646), http://tools.ietf.org/html/rfc5646

[4] World Wide Web Consortium, 2009, *Namespaces in XML 1.0 (Third Edition)*, http://www.w3.org/TR/REC-xml-names/

[5] World Wide Web Consortium, 2010, *CURIE Syntax 1.0 — A syntax for expressing Compact URIs*, http://www.w3.org/TR/curie/

[6] World Wide Web Consortium, 2013, *Turtle — Terse RDF Triple Language*, http://www.w3.org/TR/turtle/

[7] Internet Engineering Task Force, 2005, *Uniform Resource Identifier (URI): Generic Syntax* (RFC 3986), http://www.ietf.org/rfc/rfc3986

[8] Internet Engineering Task Force, 2005, *A Universally Unique Identifier (UUID) URN Namespace* (RFC 4122), http://www.ietf.org/rfc/rfc4122

[9] New Zealand Society of Genealogists, (undated), *Tags in the GEDCOM 5.5 Standard*, http://www.gencom.org.nz/GEDCOM_tags.html [accessed: 30 May 2015]

[10] Church of Jesus Christ of Latter-day Saints, 1996, *The GEDCOM Standard (Draft Release 5.5.1)*, http://wiki.webtrees.net/w/images-en/Ged551-5.pdf