GWD-C
GFSG

Freek Dijkstra, SARA
Richard Hughes-Jones, DANTE
Gregory B. Newby, Arctic Region Supercomputing Center
Joel Replogle, Open Grid Forum

May 2011

Procedure for Registration of Subnamespace Identifiers in the URN:OGF Hierarchy

Status of This Document

Community Practice

Copyright Notice

Abstract

URNs in the OGF namespace take the form
`urn:ogf:<snid>:<subnamespace-specific-string>`.

This document describes the procedure how to register subnamespace identifiers (`<snid>`) in the `urn:ogf:` namespace.

Contents

1 introduction

Uniform Resource Names (URNs) are persistent, globally unique identifiers [RFC 2141, RFC 3406].

An identifiers labels a resource, which facilitates unambiguous identification of that resource by different entities. Any resource can be named, including a work, an instance of a work, an entry in a ontology or the ontology as a whole.

1.1 Notational Conventions

The keywords "MUST" "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119].

1.2 Globally Uniqueness of URNs

[RFC 3406] stipulates:

- A URN MUST NOT be assigned to more than one resource.
- A URN MUST NOT be re-assigned to a different resource.
- A single resource MAY have more than one URN assigned to it for different purposes.

For the purpose of this document, we will call an identifier *unique* if adheres to the first two requirements above. So a unique identifier identifies only one resource, although one resource does not have to be represented by a single identifier.

Any organisation that assigns identifiers to its resources must adhere to the above requirements. This ensures that the identifiers assigned by the organisation are unique. However, this only ensures that the identifier is locally unique, but does not yet guarantee that the identifier is globally unique.

To ensure global uniqueness of identifiers, each local identifier is prepended with a *prefix* which in itself is a unique identifier of the organisation that assigns the local identifiers. In the context of URNs, this prefix is often referred to as a *namespace identifier*. The group of all identifiers that start with a certain prefix is called a *namespace*. A second purpose of a prefix is to provide a cue for the type of resource being identified.

For example, a working group in the OGF may define an ontology with three terms, identified by their names *Grid*, *Cloud*, and *Cluster*. A different organisation may use the same terms, but with different meaning. To prevent ambiguity when using these terms, the working group may register the namespace identifier `urn:ogf:example-wg`, and use the identifiers `urn:ogf:example-wg:grid`, `urn:ogf:example-wg:cloud` and `urn:ogf:example-wg:cluster`. When these identifiers are used, it is unambiguous which concept is being referred to.

This document describes the procedure to register a namespace identifier within the `urn:ogf:` namespace.

1.3 Persistency of URNs

The goal of URNs is to provide persistent identification of resources. An identifier must remain valid and unmodified from its creation to well beyond the lifespan of the resource it identifies. This persistency is the responsibility of the organisation maintaining the URN namespace as well as the delegate organisation(s) that assign the identifier.

The requirement that no URN can be re-assigned only partly ensures longevity. An identifier MUST NOT change name, and (where applicable) validation and resolution procedures SHOULD still yield results, even after the following events:

- the resource ceases to exist;
- the properties of the resources changes;
- a new version of the resource is created;
- the namespace organisation ceases to exist;
- the organisation that assigned the identifier ceases to exist; or
- the organisation that assigned the identifier changes its name.

For example, the Open Grid Forum (OGF) emerged from the merger of the Global Grid Forum (GGF) and Enterprise Grid Alliance (EGA) in 1996. Namespaces created before that time still have `ggf` in their prefix, and this MUST NOT change, as not to break the persistency of the identifier.

2 Selecting a Namespace

OGF working groups or interested third parties that wish to assign identifiers in a namespace must select a prefix. This document describes how to register a prefix in the `urn:ogf:` namespace. Alternative namespaces include:

- `urn:ogf:` namespace, as described in this document.

- `urn:` namespace, as maintained by the IETF and IANA.
- `http://schemas.ogf.org/` namespace (a location identifier that serves as a resource identifier).
- Prefix in the handle system [RFC 3650].

URNs in the `urn:ogf:` namespace are suited for identifying individual grid resources.

URNs in the `urn` namespace are suited for identifying resources other then grid resources.

A URL may be suitable as identifier for ontologies and schemata (even though a URL only identifies the location of a resource, not the resource itself), as the URL can double as the location where the normative schemata definition can be found. URLs are not suitable to identify individual resources, as the same may be available from multiple locations, or may not be accessible via a HTTP mechanism in the first place, or may outlive the lifetime of the HTTP schema (e.g. when the resource location is changed to use HTTPS).

Systems that like to integrate with the handle system for resource resolution of digital resources may consider using the identifiers used in the handle system.

DNS-based identifiers are generally not recommended for persistent identifiers.

3 Canonical Syntax of URN:OGF identifiers

The canonical syntax of URNs in the `urn:ogf:` namespace is specified in section 2.4 of [draft-dijkstra-urn-ogf]. This section is replicated here.

```
OGF-URN = "urn:ogf:" SNID ":" SUBNAMESPACE-SPECIFIC-STRING
```

where `SNID` is a unique subnamespace identifier, and `SUBNAMESPACE-SPECIFIC-STRING` is a unique local identifier within the namespace with the prefix `"urn:ogf:" SNID ":"`.

`SNID` has the same syntax as a `NID` as defined in [RFC 2141]:

```
SNID = ( ALPHA / DIGIT ) *31( ALPHA / DIGIT / "-" )
```

`ALPHA` and `DIGIT` are defined in Appendix B of [RFC 5234].

Both `"urn:ogf:"` and the subnamespace identifiers are case insensitive.

The next section describes the procedure how subnamespace identifiers are assigned by the OGF.

The syntax of `SUBNAMESPACE-SPECIFIC-STRING` is dependent on the `SNID`. [draft-dijkstra-urn-ogf] does not pose any additional restrictions to the `SUBNAMESPACE-SPECIFIC-STRING` other than what is defined in the NSS syntax as defined by [RFC 2141] or its successor:

```
SUBNAMESPACE-SPECIFIC-STRING = 1*<URN chars>
```

<URN chars> is defined in Section 2.2 of [RFC 2141].

At the moment of writing, a revision of [RFC 2141] is proposed, which may allow a different set of allowed characters (URN-char) in the NSS than the current URN chars (adding "&" and/or "~", whilst removing the reserved characters "%", "/", "?" and "#") [RFC2141bis]. The intention of this document is to only restrict the syntax of the SNID, and have Grid Forum Documents specify the syntax of the SUBNAMESPACE-SPECIFIC-STRING. It is RECOMMENDED that these Grid Forum Documents explicitly list the allowed characters.

4 Procedure for Registering a Namespace Identifier

The Open Grid Forum delegates part of their namespace urn:ogf: by assigning subnamespace identifiers (SNIDs). The formal application for a SNID is made via publication of an Grid Forum Document (GFD). The GFD should be an Information Document, and should be published according to the normal procedure as described in [GFD-C.152] or its successor.

A GFD that registers a namespace MUST include the following sections:

- The prefix to register
- The version of the specification (usually starting with version 1)
- Syntax of the namespace
- Rules for lexical equivalence of two URNs in the namespace
- Procedure for assignment of identifiers (and possible subdelegation of the namespace)
- Identifier persistence considerations

The template defined in section 5 MAY be used as part of an GFD that registers a namespace.

The Open Grid Forum maintains a registry of namespace identifiers in the urn:ogf: namespace at the URL http://www.ogf.org/urn/.

It is RECOMMENDED to notify the technical director of the OGF of any (proposed) use of a namespace identifier, by contacting urn@ogf.org. The technical director of the OGF may, at his discretion "reserve" a subnamespace identifier for a period of maximal one year prior to approval of an Informational GFD document describing the namespace.

This document does not describe the resolution procedure of any namespace conflicts.

5 Template for Registering a Namespace Identifier

1. Registration

1.1. Namespace Identifier

The subnamespace identifier.

> Example: “urn:ogf:example:”

Consideration: This namespace is likely to exist for a prolonged period of time. Do not include volatile data, such as a working group name (example-wg), although a name of a protocol is acceptable (e.g. example for the example-wg working group). urn:ogf: is a formal namespace, subnamespace identifiers MUST NOT start with x-.

1.2. Document Version

Registration version number: starting with 1, incrementing by 1 with each new version.
Registration date: date of submission of the GFD.

1.3. Declared Registrant

The name and address of the namespace organisation and/or contact person that is responsible for making the registration.

This section may be omitted if the registrant is the same as the authors of the GFD.

2. Syntax

2.1. Syntactic Structure

This section should outline any structural features of identifiers in this namespace. Is should describe the syntax of all current and future URNs within this namespace that are considered valid. This description may be used to introduce terminology used in other sections.

> Example:

```
EXAMPLE-URN = "urn:ogf:" SNID ":" LOCODE ":" SUPERCOMPUTER
         *1(QUERY) *1(FRAGMENT)
SNID = "example"
LOCODE = CC "-" CITY ; UN location code, http://www.unece.org/locode/
CC = 2( ALPHA ) ; 2-letter country code
CITY = 3( ALPHA / DIGIT ) ; UN/LOCODE can contain A-Z and 2-9.
```

```
SUPERCOMPUTER = 1*<URN chars> ; Identifier of a historic supercomputer
QUERY = "?" 1*<URN chars>
FRAGMENT = "#" 1*<URN chars>
```

Example-URNs that are assigned to a resource MUST contain a UN/LOCODE that is valid at the time of assignment. URNs must not change, even not when a UN/LOCODE is changed, and those URNs are still considered valid. Clients MAY infer from the UN/LOCODE that the supercomputer is located at the designated location.

The supercomputer string may be used for display purposes, but it is opaque: clients MUST NOT infer any attributes about the resource being identified for the `SUPERCOMPUTER`.

Consideration: If part of the structure if opaque in meaning (the meaning is not exposed), this should be noted. The specification may use the augmented BFR format [RFC 5234].

This section may be combined with section 2.2 or 2.3.

2.2. Reserved Use

This section should outline which of valid URNs MAY be assigned, and which URNs SHOULD NOT (yet) be assigned, but are reserved for future use.

Example: For current assignments, `SUPERCOMPUTER` MUST only contain the characters: `SUPERCOMPUTER = 1*(ALPHA / DIGIT / "(" / ")" / "+" / "," / "-" / "." / "=" / "@" / ";" / "$" / "_" / "!" / "*" / "’" / "%" <hex> <hex>).`

The `<URN chars>` may later be expanded with the “`&`” and “`~`” characters [RFC2141bis]. These characters MUST NOT be used for assignments, but parsers SHOULD be prepared to accept URNs that contain these characters, as well as the characters “`:`” “`~`” “`&`” and “`/`”.

The `QUERY` and `FRAGMENT` part MUST NOT be included in assigned URNs, until it use is documented by a revision of this document.

Percentage-escaped strings MAY be assigned, but SHOULD NOT contain upper case characters, nor combining diacritical marks, nor ligatures. Receiving software MUST accept these characters. This will avoid confusion for user who otherwise may assume that the encoded characters in a URN are also case insensitive (they are not), but still allow future inclusion of these characters.

Considerations: [RFC 2141] does not allow query or fragment identifiers, but the they may be allowed in the future. It is recommended to explicitly note if query or fragment identifiers are allowed in the namespace.

This section may be combined with section 2.1 or 2.3.

2.3. Encoding

This section should specify the encoding or decoding algorithms. This is particularly applicable in the case (a part of) a URN is maps to a legacy naming systems.

This section MUST specify if percent-escaping as defined in [RFC 2141] is allowed, and if so, what the binary code represents. This section MUST specify if the `<reserved>` characters as defined in `<URN chars>` in [RFC 2141] are allowed (it is recommended that they are not).

> Example:
>
> - The `LOCODE` part MUST map to a valid UN/LOCODE string, with the `" "` (space) in the UN/LOCODE mapped to a `"-"` in the LOCODE part of the URN. Since no `"-"` can occur in the UN/LOCODE, this mapping can be inverted.
> - The `SUPERCOMPUTER` part MAY contain percent-escaped bytes (`"%" <hex> <hex>`) as described in [RFC 2141]. The percentage-decoded `SUPERCOMPUTER` MUST be a valid UTF-8 encoded byte sequence.
> - Assigning organisations SHOULD only assign `SUPERCOMPUTER` whose Unicode code points are NFKD-normalised and are allowed by the rules stipulated in [RFC 5892] (no upper case or control code points).
> - Receiving software SHOULD accept URNs that are valid UTF-8 encoded, even if it contains disallowed code points. Receiving software MUST NOT display the decoded URN if it contains disallowed code points.
>
> For example the `SUPERCOMPUTER` string `"Colossus%20Mark%20I"` is allowed ("Colossus Mark I"); `"Colossus%20Mark%20%E2%85%A0` is not valid. `"%E2%85%A0"` is a valid UTF-8 representation of the code point Roman Numeral I, but this code point is not allowed by [RFC 5892], nor is it properly NKFD-normalised; `"J%C3%9CGENE""` is valid. `"J%c3%bcgene""` is not valid. While it is valid UTF-8 and all code points are allowed by [RFC 5892], it is not normalised with NFKD;

This section may be combined with section 2.1 or 2.2.

2.4. Rules for Lexical Equivalence

This section should list the algorithm for determining lexical equivalence between two identifiers in the underlying namespace.

> Example: URNs are lexical equivalent if and only if they are byte-equivalent after case normalisation.
>
> No interpretation or normalisation of percentage-escaped characters should take place.

Considerations: [RFC 2141, draft-dijkstra-urn-ogf] specify that the `"urn:ogf:"` part and the SNID are case insensitive and that hex-encoding in a percentage-escaped character is case insensitive. The lexical equivalence algorithm defined in this section MUST NOT violate these requirements. Typical normalisation functions to consider are case normalisation, percentage-encoding, diacritical normalisation using canonical (de)composition (e.g. NFC and NFD in Unicode), and ligature normalisation using compatibility (de)composition (e.g. NFKC and NFKD in Unicode). If Unicode normalisation is to occur, the specification should list a specific Unicode version (e.g. Unicode 2.0 or Unicode 6.0) since this function may change between versions (e.g. there is no pre-composed character for Y-caron, but this may be created in future Unicode version).

It is recommended to list a few examples.

> Example: Consider the following URNs (`"Colossus%20Mark%20%E2%85%A0"` decodes to "*Colossus Mark I*"):
>
> 1- `urn:ogf:example:GB-BLE:Colossus%20Mark%20I`
> 2- `urn:ogf:example:gb-ble:colossus%20mark%20i`
> 3- `urn:ogf:example:gb-ble:%63olossus%20mark%20i` ; `0x63` is "c"
> 4- `urn:ogf:example:GB-BLE:%43olossus%20Mark%20I` ; `0x43` is "C"
>
> Only URN 1 and URN 2 are equivalent to each other.
>
> Note that URN 4 contains a code point (0x43) currently SHOULD NOT be assigned, but MUST be accepted once received.

3. Services

3.1. Validation mechanism

A URN namespace may provide mechanisms for "validating" a URN – i.e., determining whether a given string is currently a validly-assigned URN. For example, even if a telephone number-based URN namespace was created, it is not clear that all telephone numbers would immediately become "valid" URNs, that could be resolved using whatever mechanisms are described as part of the namespace registration.

Where applicable, this section should describe if the operation of validation servers is an open process, or that it is subject to some authoritative delegation procedure.

Considerations: If such mechanism exists, this section may describe this mechanism, or may refer to a document that does describe it. This section may contain requirement how a recipient of a malformed URN should act: accept the URN as-is, attempt to normalise it or reject it.

This section is optional.

3.2. Process for identifier resolution

A URN namespace may provide mechanisms for “resolving” a URN – i.e., determining attributes of the resources that is described, such as metadata or the location(s) where the resource can be retrieved.

Where applicable, this section should describe if the operation of resolution servers is an open process, or that it is subject to some authoritative delegation procedure.

Considerations: If such mechanism exists, this section may describe this mechanism, or may refer to a document that does describe it. This section may contain an estimate of the volatility of the resource attributes and a reasonable caching time for these attributes, or it could dictate that any resolution mechanism contains caching time-outs.

This section is optional.

4. Namespace Considerations

4.1. Scope

This section should outline the scope of the use of the identifiers in this namespace. Apart from considerations of private vs. public namespaces. It is recommended that a namespace is limited in scope, to enhance its quality. For example, a namespace claiming to deal in “computers” should have a global scope and address all computers, which is unlikely. On the other hand, a namespace claiming to deal with only historic supercomputers is more reasonable.

Note: It is expected that more than one namespace may serve the same “functional” purpose; the intent of the “Namespace Considerations” section is to provide a record of the proposer’s “due diligence” in exploring existing possibilities, for the IESG’s consideration.

> Example: The example URN namespace only deals with historic computers that are considered landmark achievements in computer science. It is limited to supercomputers that are immobile due to their sheer size.

This section may be combined with section 4.2.

4.2. Identifier uniqueness considerations

This section should address the requirement that URN identifiers be assigned uniquely – they are assigned to at most one resource, and are not reassigned.

This section should address the type of resource to be identified. The definition of resource is fairly broad. For example, for bibliographic records the following resource types exists:

work e.g. Jane Austin's "Gone with the Wind", or the

expression e.g. the Swedish translation of the work

manifestation e.g. a hard cover of the expression

item e.g. an actual book on a shelf

In particular, this section should specify if it deals with a general resource (e.g. "the weather in Seattle"), a particular manifestation or version (e.g. "the weather in Seattle on May 1st, 2011") or a dynamic manifestation (e.g. "the current weather in Seattle"). If desired, the URN syntax may include a part which specifies the version of the resource (e.g. `urn:ogf:example:1943-12-08:GB-BLE:Colossus%20Mark%20%E2%85%A0`)

> Example: Each URN refers to a particular instance of the computer in it's final state. Replicas and revisions of the original design should receive their own URN.

This section may be combined with section 4.1 or 4.3.

4.3. Identifier uniqueness considerations

This section should include any other considerations dealing with the uniqueness of URNs in the namespace. Possible considerations include, but are not limited to:

- exposition of the structure of the identifiers, and partitioning of the space of identifiers amongst assignment authorities which are individually responsible for respecting uniqueness rules
- identifiers are assigned sequentially
- information is withheld; the namespace is opaque

This section may be combined with section 4.2.

5. Community Considerations

5.1. Process of identifier assignment

This section should detail the mechanisms and/or authorities for assigning URNs to resources. It should make clear whether assignment is completely open, or if limited, how to become an assigner of identifiers, and/or get one assigned by existing assignment authorities.

Answers could include, but are not limited to:

- assignment is completely open, following a particular algorithm
- assignment is delegated to authorities recognised by a particular organisation
- assignment is handled on a per-identifier basis by community consensus (e.g. by publication of a community reviewed Grid Forum Document.)

5.2. Identifier persistence considerations

The non-reassignment requirement of URN identifiers ensures the persistency of a URN after the lifetime of a resource. Despite good intentions, there may still be valid reasons for a desire to change a URN identifier:

- The prefix contains the name of assigning organisation, and the assigning organisation merges, ceases to exists or changes name.
- The identifier contains certain attributes of the resource, and these attributes change.
- The resource changes

For example, the identifier `urn:ogf:example:GB-BLE:Colossus%20Mark%20%E2%85%A0` contains the location "Bletchley" in "Great Britain", as well as the name "Colossus Mark I" of the resource. GB could be changed to UK; the community of Bletchley could be annexed by the city of London; it is discovered that the name of the resource is misspelled; a marketing decision was made to change the name, etcetera. While these events rarely occur, their occurrence may still be larger than the anticipated lifetime of the resource identifier, and they must therefore still be considered.

Considerations: This section should include the considerations why this particular syntax of the URN is most likely to remain stable, even in the case of rare events. It may make a realistic estimate of the lifetime of the resource, a maximum lifetime of the resource *identifier*, and from that deduce a minimum time that the resource identifier may not be re-allocated. This section may explain the organisational requirements how this minimum time is ensured.

> Example: The Example-URN namespace uses historic names of (super)computers. It is expected that these names rarely change. No other attributes are part of the name. To avoid name conflicts, the identifier also includes the location of the machine, assuming that these machines rarely move location. The location name at time of commissioning is used, to avoid renaming issues when the location changes name. If the computer is moved location, either the old identifier is retained, meaning that the location only infers the name of the original location, not the name of the current location. It was anticipated that the location is more stable than the name of the commissioning organisation. The lifespan of these identifiers may be centuries (e.g. to describe Charles Babbage's *difference engine*.

All identifiers must be assigned by a central authority, to ensure longtime preservation of the identifiers.)

6. Examples

This section may list a few example identifiers.

Example: The following examples of Example-URNs are informative only. They may not actually exist:

```
urn:ogf:example:GB-BLE:Colossus%20Mark%20%E2%85%A0
urn:ogf:example:US-APG:ENIAC
urn:ogf:example:JP-YOK:Earth%20Simulator
urn:ogf:example:US-YKH:Watson
```

7. Relevant Ancillary Documentation

This section should list any GFDs, RFCs, standards, or other published documentation that defines or explains all or part of the namespace structure.

This section is optional and may be replaced with in-line bibliography references.

8. Changes from Previous Versions / Prior Usage

This section should include changes to previous versions of the document, or list prior (undocumented) use of the namespace.

This section is optional.

6 Security Considerations

There are no additional security considerations other than those normally associated with the use and resolution of URNs in general.

Implementors are recommended to check the OGF registry and documentation at http://www.ogf.org/urn/ before assuming that a given identifier is valid or has a certain meaning.

7 Glossary

Assigning Organisation A namespace organisation that assigns unique identifiers to resources.

Cool URI A URL (!) that is persistent and is used as both a locator and an identifier.

Fragment Identifier The part of a URN that follows a pound sign ("#").

Globally Unique Identifier An identifier which identifies at most one resource worldwide. A global identifier typically consists of a prefix and a local identifier.

Identifier A label (string or byte sequence) that identifies a resource.

Locally Unique Identifier An identifier assigned by an assigning organisation, that identifies at most than one resource.

Namespace A set of all possible identifiers with the same namespace identifier.

Namespace Identifier A unique identifier of a namespace organisation, often used as a prefix.

Namespace Organisation An organisation maintaining a namespace. A namespace organisation may delegate part of their namespace to another namespace organisation.

Prefix A namespace identifier, which is prepended to a local identifier to form a globally unique identifier.

Question Identifier The part of a URN that follows a question mark ("?").

Subnamespace Identifier A word that identifies a part of a a larger namespace. The string "urn:ogf:" followed by a subnamespace identifier forms the prefix of a namespace that is subject of the registration procedure as described in this document.

Unique Identifier An identifier that identifies at most one resource, and is never reassigned.

URL The location where a resource can be found.

URN A globally unique identifier that adheres to the URN syntax described in [RFC 2141].

8 Contributors

Freek Dijkstra (Editor)
SARA
Science Park 121

1098 XG Amsterdam
The Netherlands
Email: Freek.Dijkstra@sara.nl

Richard Hughes-Jones
DANTE
City House
126-130 Hills Road
Cambridge CB2 1PQ
United Kingdom
Email: Richard.Hughes-Jones@dante.net

Gregory B. Newby
Arctic Region Supercomputing Center
P.O. Box 756020
Fairbanks, Alaska 99775
USA
Email: newby@arsc.edu

Joel Replogle (Corresponding Author)
Open Grid Forum Office
P.O. Box 2326
Joliet, Illinois 60434
USA
Email: replogle@ogf.org

9 Acknowledgments

This document is based on the best practice of delegation in the URN hierarchy, as described in RFC 3406.

Intellectual Property Statement

can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF 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.

Full Copyright Notice

References

[draft-dijkstra-urn-ogf] Freek Dijkstra and Richard Highes-Jones. A URN Namespace for the Open Grid Forum (OGF). draft-dijkstra-urn-ogf (Informational), April 2011. URL `http://tools.ietf.org/html/draft-dijkstra-urn-ogf`.

[GFD-C.152] Charlie Catlett, Cees de Laat, David Martin, Gregory B. Newby and Dane Skow. Open Grid Forum Document Process and Requirements. GFD-C.152 (Community Practice), 24 June 2009. URL `http://www.ogf.org/documents/GFD.152.pdf`.

[RFC 2119] Scott Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (Best Current Practice), March 1997. URL http://tools.ietf.org/html/rfc2119.

[RFC 2141] Ryan Moats. URN Syntax. RFC 2141 (Standards Track), May 1997. URL http://tools.ietf.org/html/rfc214.

[RFC2141bis] Alfred Hoenes (Editor). Uniform Resource Name (URN) Syntax. draft-ah-rfc2141bis-urn (Standards Track, if aproved), May 2010. URL http://tools.ietf.org/html/draft-ah-rfc2141bis-urn.

[RFC 3650] Sam Sun, Larry Lannom and Brian Boesch. Handle System Overview RFC 3650 (Informational), August 2002. URL http://tools.ietf.org/html/rfc3305.

[RFC 3406] Leslie L. Daigle, Dirk-Willem van Gulik, Renato Iannella and Patrik Fältström. Uniform Resource Names (URN) Namespace Definition Mechanisms. RFC 3406 (Best Current Practice), October 2002. URL http://tools.ietf.org/html/rfc3406.

[RFC 5234] Dave Crocker and Paul Overell. Augmented BNF for Syntax Specifications: ABNF. RFC 5234 (Standards Track), January 2008. URL http://tools.ietf.org/html/rfc5234.

[RFC 5892] Patrik Fältström (editor). The Unicode Code Points and Internationalized Domain Names for Applications (IDNA). RFC 5892 (Standards Track), August 2010. URL http://tools.ietf.org/html/rfc5892.