



USGIN

U.S. Geoscience Information Network

USGIN URI Policies

Version 1.1

Title:

USGIN URI Policies

Latest released version:

<http://repository.usgin.org/uri-gin/usgin/dlio/331>

Creator:

USGIN Standards and Protocols Drafting Team

Editors:

Stephen M. Richard

Creation date:

2/26/2010

Last revision date:

8/25/2011 5:28 PM

Document Status:

Release V. 1.1

Publisher:

Arizona Geological Survey

Description:

This document presents a proposal for minting of dereferenceable http URIs for use in interoperable web services for the U. S. Geoscience Information Network

Identifier:

USGIN2011-001

Notices

Neither the USGIN project, nor any of the participating agencies, take any position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither do they represent that there has been any effort to identify any such rights.

This document and the information contained herein is provided on an "AS IS" basis and USGIN 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.

DRAFT

Revision History

Version	Date	Comments	By
1.0.0	2010-02-26	Initial document	Stephen Richard
1.0.1	2010-02-26	Formatting, Word & PDF preparation for public release	Wolfgang Grunberg
1.0.3	2010-03-15	Add discussion based on UK public sector URI scheme, other edits	Stephen Richard
1.1	2011-08-25	update logo on title page, add identifiers. Change uri-gin to uri-gin in URI syntax. For posting in repository.usgin.org; original release at http://lab.usgin.org/node/315	Stephen Richard

DRAFT

Table of Contents

Summary.....	5
Introduction.....	5
URI requirements	7
Issues related to identifier syntax	7
Issues related to community practice	7
Issues related to stewardship.....	8
Identifier scheme syntax	8
Generic URI scheme	8
http URI scheme	9
USGIN URI syntax.....	9
Interpretation of a USGIN URI	10
Identifier equivalence.....	12
Fragments.....	12
Versions	13
Canonical forms.....	13
Functional Resources.....	13
Meta-resource identifiers and registry structure.....	13
Comparison with other schemes.....	14
UK public sector.....	14
Examples.....	14
Special URIs.....	14
URI host	15
URI profile.....	15
Naming authority	15
Resource URIs using the uri-gin profile	15
Person.....	15
Vocabulary (codelist).....	15
Term, concept, classifier.....	15
Organization	16
Map	16
Dataset	16
Feature	17
Document.....	17
Web application	17
Application.....	17
Service	17
Mapping to CGI URN	18
Resource Types.....	19
References.....	28
Appendix 1. Collected ABNF for URI, RFC-3986	28
Appendix 2. Collected ABNF for URI, RFC-2396	30
File name restrictions	32

Summary

This document presents a proposal for minting of dereferenceable http URIs for use in interoperable web services. The design approach is to explicitly distinguish the dereferencing host from the name authority, recognizing that these may be decoupled. Content negotiation and url redirection that are invisible to the user violate the requirement for URI transparency. One of the objectives of this URI scheme is to define a syntax that makes clear the distinction between an identifier for a non-information resource, information resource and specific representation resource to address the issues raised by Booth (2003-01-28). Another important design criterion is that the http URIs should be dereferenceable using standard web server resource path conventions on any commonly used operating system, in order to make implementation of dereferencing as simple as possible. The ABNF syntax can be summarized as:

```
ginURI = "http:" "/" uriHost "/" URIScheme "/" nameAuthority "/"  
resourcePath [ "/" resourceSpecificString ] [ ( "/" / "/" representationPart ) ]
```

Introduction

In the World Wide Web, identifiers are bit streams. They only have meaning in the mapping between the identifier bit stream and an identified thing. In natural language these identifiers are the words used for communication, but natural language is far too ambiguous for use in machine processing. The view taken here is that to be useful in an information system, an identifier bit stream should correspond to exactly one identified thing. Any thing that can be the subject of communication can have an identifier. For the meaning of the identifier to be discovered, there must be a dereferencing mechanism that yields a description of the identified thing in a form useful to the requesting agent. In this discussion the term resource is used to mean an identifiable thing that is useful for some purpose.

There are two facets of a resource that are important for scoping identifiers -- information resource vs. non-information resource, and concrete vs. abstract resource.

An information resource is something that can be transmitted electronically. Documents, such as web pages, RDF/XML documents, and binary files are all information resources. Non-information resources are those resources that can be identified by a URI but which cannot be transmitted electronically. Human beings, abstract concepts, etc. are non-information resources (c.f. Booth, 2003-01-28). Both information and non-information resources can be identified with URIs.

Resource type	notes	examples
concrete information resource	a specific representation of a resource. Equivalence is determined by bitwise matching of the representation bitstream	monaLisa.tif, USGIN_URI_scheme_1.0.1.doc
abstract information resource	an information resource defined by a particular content scope, includes most kinds of documents. Equivalence is determined by matching of content meaning or intention (depending on how the resource is defined)	Image of Mona Lisa, current version of USGIN URI scheme, the Declaration of Independence
concrete non-information resource	An identifiable physical object or event.	The original Mona Lisa painting, The original Declaration of Independence parchment, a particular rock sample, a particular car, a particular meeting, a particular project.
abstract non-information resource	A concept defined by some human intention, may categorize other resources, or represent some abstract idea.	painting, love, the US Congress

The importance of this distinction for use in the World Wide Web is that an information resource is defined such that the resource itself (a normative representation) can be transmitted electronically. There may be multiple representations of an information resource (file formats), but one of these can be defined to be the normative representation.

A computer can only display information resources, but they are (by definition) incapable of displaying a non-information resource itself (the normative form exists outside of the computer, or is abstract). Non-information resources can only be presented electronically using a representation, which is a concrete information resource meant to communicate the nature of the non-information resource that corresponds to the URI. Examples of representations include pictures of people, a free text description, or a description using formal syntax like xml or OWL. Some non-information resources are defined functionally (a web service, a software application), in which case the information-resource representations are software files that implement the functionality.

The dereferencing of information and non-information resources is fundamentally different. Dereferencing a URI for information resource should present the client with a normative representation (canonical form) that is the actual resource. Dereferencing a URI that represents a non-information resource returns a representation of the resource. There may be many valid representations of non-information resources, useful in different contexts. Each of these representations is itself an identifiable information resource and the canonical or default representation may be context dependent.

Some approach to bundling the identifier with a dereferencing protocol is essential, and http + DNS provide it (Fielding et al., 1999-06). The W3C Technical Architecture group has determined that if an HTTP response code of 200 (a successful retrieval) is given, then the URI identifies an information resource, but with no such response, or with a different code, no such assumption could be made. (See <http://www.w3.org/2001/tag/issues.html#httpRange-14>; Berners-Lee, 2005-06-09, <http://www.w3.org/DesignIssues/-HTTP-URI2>; <http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14>, from which all content was subsequently removed (2007-10-4)). The issue with such a model is that if a 300 code response results in a redirect this distinction will be invisible to a human user, and also precludes the use of redirects to handle relocation of the dereferencing host for legacy URIs. This is not accepted as a viable solution to meet the requirements; our assertion is that the best solution is to explicitly distinguish the dereferencing host from the name authority, recognizing that these may be decoupled. Content negotiation and URL redirection that are invisible to the user violate the requirement for URI transparency (see below).

If http URIs are to be used for identifying information and non-information resources, and the URIs are expected to be dereferenceable using the existing DNS system on the Internet, careful consideration is necessary to clarify what is actually identified. As pointed out by Booth (2003-01-28) there are at least four distinct but related resources that we might like to identify using an HTTP URI, e.g. "http://resources.usgin.org/granite". These are:

1. The actual string that is contained in the quotes. This will be referred to here as the identifier label (Booth uses 'name'), and is an information resource. This is not an interesting case for the requirements outlined below. Enclosing the string in quotes will be used to indicate that the reference is to the actual sequence of characters (independent of the character encoding used).
2. The concept of granite, a non-information resource.
3. A 'Web Location', which can be thought of as the information resource that is produced by an HTTP GET request using that URI. There is no guarantee that the same web location will GET the same document instance when it is recalled. As such, a Web Location identifies an abstraction. An organization's home page is a typical example. An analogous resource would be a document in a computer file system that may be edited and change over time, or converted to a different format, but is considered to be the same document because of its intended content and purpose. A web location may also dereference to an application that delivers some particular functionality, or may be a base URL for a web service. In these cases the http URI identifies a software application.

4. A particular document instance, characterized by its content and format. This information resource is analogous to a particular file at a particular time on a computer hard drive, and identity is based on equivalence of the bitstream resulting from an HTTP GET with the identifier.

URI requirements

This section summarizes some requirements for URIs in the USGIN system. The citations are meant to recognize the sources from which the requirements were acquired, and do not imply endorsement of their inclusion in this list by the authors, or that the sources are normative. The philosophical approach is to create URIs that are intended for use in interoperable web services by people, and that simplify implementation of dereferencing services.

Issues related to identifier syntax

Identifiable -- Identifiability of a URI means that the agent responsible for the identified resource is communicated to a person seeing the URI. URI identifiability is a form of advertising, where the admittedly modest impact of a single use of an identifiable URI is potentially magnified greatly by widespread replication. The desire for branding to be evident in URIs is both widespread and understandable. Identifiability also is a cornerstone of trust: brand recognition and successful URI access are mutually reinforcing. (Thompson and Rees, 2009). This requirement contrasts with the recommendation that UK public sector URIs should not contain the name of the department or agency responsible for it (Davidson, 2009-10). That condition makes sense for URIs scoped to UK public sector, but in a broader community like USGIN with a variety of independent naming authorities, the benefit of identifying the name authority outweighs the cost.

Transparent -- It should be evident what a URI is about by inspection. This requires documenting the nature of the mapping from URIs to resources as part of the specification of a URI scheme (Thompson and Rees, 2009). In particular, it should be apparent from the form of a URI whether it identifies an information resource or non-information resource. It should be possible to determine the identifier for a resource from an identifier for a representation of that resource. Opaque MIME format negotiation between client and server is not a good idea-- it's too easy to confuse people about what the URI represents if the same identifier produces different representations in different circumstances. Communities of practice could agree on a canonical form that is the default representation, otherwise representation type/format should be explicit in the URI.

Memorable -- A URI often has to be remembered by people, and it is easier for people to remember a URI when it consists of meaningful or familiar components. (Berners-Lee et al., 2005-01)

Keyboard compatible -- A URI might be transcribed from a non-network source and thus should consist of characters that are most likely able to be entered into a computer, within the constraints imposed by keyboards (and related input devices) across languages and locales. (Berners-Lee et al., 2005-01)

Portable -- Should be able to change dereferencing host system without reengineering identifiers; decouple locator function from identification function (USGIN)

Compatible with network paths -- The syntax used in URIs should be compatible with file system paths used by common operating systems (Unix, Linux, MacOS, MS Windows), and should be easily mapped to file paths that a web server uses to identify information resources that will be returned by HTTP GET requests. The objective is to make implementation of URI dereferencing simple, ideally using the functional capabilities in standard off the shelf software like web browsers, with a minimum of special software components (USGIN).

Issues related to community practice

Useable -- It should require little or no effort on the part of ordinary users to retrieve a useful representation of the resource identified by a URI in the scheme (Thompson and Rees, 2009).

Self-describing -- Given a URI in some known scheme it should be possible to retrieve metadata about the URI and the resource it identifies independently of the representation of that resource. (Thompson and Rees, 2009)

Reliable -- It should always be possible to get a positive response (either a representation or other definite advice about the resource) from an attempt to dereference a URI in the scheme. (Thompson and Rees, 2009)

Documented -- Naming authorities should make resources available that document how they are using a URI scheme. Canonical representations of non-information resources should be well defined (USGIN).

Issues related to stewardship

Distributed -- the owner of a set of URIs in the scheme must be able to delegate support for the transfer of naming authority (control over the meaning of URIs) for designated parts of the scheme. (Thompson and Rees, 2009)

Owner stability -- ownership of a URI, and the authority over a URI's meaning which follows from it, should continue as long as the owner wants it to (Thompson and Rees, 2009)

Resource stability --The resource that a URI identifies should not change (Thompson and Rees, 2009). If a resource is evolving through time, a system for versioning the resource should be encoded in the URI such that different versions may be distinguished (USGIN).

Cost -- complex schemes requiring complex dereferencing software cost more to implement and maintain than schemes for which existing, widely deployed, off the shelf software can be used for dereferencing.

Identifier scheme syntax

This section provides an abbreviated overview of the IETF specification for URI syntax in the World Wide Web (IETF RFC3986), and the specification of the more specific http URI scheme defined for use with hypertext transfer protocol (HTTP, RFC2616), which is based on now superseded RFC2396 URI syntax. Skip this section if you are well versed in the formal syntax for generic URI and http URI.

Generic URI scheme

The major ABNF rules that define the generic URI scheme are summarized here from RFC3986 (<http://tools.ietf.org/html/rfc3986>) (which superseded RFC2396). ABNF notation is described in Crocker and Overell (2008-01). The key rules include:

```
URI = scheme ":" hier-part [ "?" query ] [ "#" fragment ]
hier-part = "//" authority path-abempty ; path begins with "/" or is empty
           / path-absolute ; begins with "/" but not "//"
           / path-rootless ; begins with a segment
           / path-empty ; zero characters
authority = [ userinfo "@" ] host [ ":" port ]
host = IP-literal / IPv4address / reg-name
reg-name = *( unreserved / pct-encoded / sub-delims )
```

The presence of a host subcomponent within a URI does not imply that the scheme requires access to the given host on the Internet. In many cases, the host syntax is used only for the sake of reusing the existing registration process created and deployed for DNS, thus obtaining a globally unique name without the cost of deploying another registry.

A host identified by a registered name is a sequence of characters usually intended for lookup within a locally defined host or service name registry, though the URI's scheme-specific semantics may require that a specific registry (or fixed name table) be used instead. For URIs intended to have global scope, a globally scoped nam-

ing system is necessary for the registered name (*reg-name*) of a host. The most common global name registry mechanism is the Domain Name System (DNS).

A registered host name that is intended for lookup in the Domain Name System (DNS) uses the syntax defined in Section 3.5 of RFC1034 (Mockapetris, 1987) and Section 2.1 of RFC1123 (Braden, 1989). Such a name consists of a sequence of domain labels separated by ".", each domain label starting and ending with an alphanumeric character and possibly also containing "-" characters. RFC3986 (Berners-Lee et al., 2005) states that URI producers should use names that conform to the DNS syntax, even when use of DNS is not immediately apparent, and should limit these names to no more than 255 characters in length.

URIs that are hierarchical in nature use the slash "/" character for separating hierarchical components. For some file systems, a "/" character (used to denote the hierarchical structure of a URI) is the delimiter used to construct a file name hierarchy, and thus the URI path will look similar to a file pathname. This does NOT imply that the resource is a file or that the URI maps to an actual file system pathname.

http URI scheme

http URI's are formally defined in RFC2616 (Fielding et al., 1999-06), which imports definitions of "URI-reference", "absoluteURI", "relativeURI", "port", "host", "abs_path", "rel_path", and "authority" defined by RFC2396, which is obsoleted by RFC3986. The following syntax rules use the Augmented Backus-Naur Form (ABNF) notation of RFC5234 (Crocker and Overell, 2008-01). See Appendix 2. Collected ABNF for URI, RFC-2396 for complete ABNF definition of elements from RFC2396.

```
http_URL = "http:" "/" host [ ":" port ] [ abs_path [ "?" query ] ]
```

In terms of RFC3986, the RFC2396 `host [":" port]` segment is the authority, and RFC2396 `abs_path` corresponds to the `path-abempty` part of the `hier-part` in a URI. The significant difference between RFC3986 and RFC2396 in this case is that the revised specification allows an empty path string, which is proscribed by RFC2396. The `host` rule is also revised in RFC3986, but because USGIN URI's do not treat the `host:port` part of the URI as part of the actual resource identifier, these do not matter (see below).

If the port is empty or not given, port 80 is assumed. The semantics are that the identified resource is located at the server listening for TCP connections on that port of that host, and the Request-URI for the resource is `abs_path` (RFC 2616, section 5.1.2). The use of IP addresses in URLs SHOULD be avoided whenever possible. If the `abs_path` is not present in the URL, it MUST be given as "/" when used as a Request-URI for a resource. The Request-URI is a Uniform Resource Identifier and identifies the resource upon which to apply the http request.

```
Request-URI = "*" | absoluteURI | abs_path | authority
```

The four options for Request-URI are dependent on the nature of the request.

Although the http URI scheme is named after the http protocol, this does not imply that use of these URIs will result in access to the resource via http.

USGIN URI syntax

URI scheme specifications must define their own syntax so that all strings matching their scheme-specific syntax will also match the RFC3986 `absolute-URI` grammar, as described in Section 4.3 of RFC3986. USGIN URI's are intended to function as locators as well, so their syntax must also conform to rules for `http_URL` as specified in RFC2616. See *Appendix 1. Collected ABNF for URI, RFC-3986* for explanation of BNF not defined here.

```
httpURI = "http:" "/" uriHost "/" URIScheme "/" identifier
```

```
ginURI = "http:" "/" uriHost "/uri-gin/" identifier
```

```

URIScheme = "uri-" safestring
uriHost = ( IP-literal / IPv4address / reg-name ) [ ":" port ]
port = *DIGIT

identifier = resourcePart [ ("/" / "/" representationPart ) ] ; for a non-
information resource, the resource part has a terminal '/'; an information
resource URI is not terminated by a '/'.

resourcePart = nameAuthority "/" resourcePath [ "/" resourceSpecificString ]
nameAuthority = safeString
resourcePath = resourceType *( "/" resourceType )
resourceType = safeString
resourceSpecificString = safeString ; Recommended practice is to use the '.' to
represent hierarchical relationships that are not reflected in the primary
hierarchy represented by the '/' delimiters. See examples. This use of '/'
vs. '.' for representing hierarchy in the URI is up to the name authority,
and may be motivated by decisions about organization of a directory-based
dereferencing implementation, a to represent semantics in the resource iden-
tification. Individual name authorities may decide to assign special seman-
tics to other special characters ("-"/"."/"_" / "~") in the unreserved
element.

representationPart = safeString [ "." safeString ] ; If a '.' is present, the
identified representation is implied to be a specific format, which may be
different from the canonical form. Note that syntactically (and conceptual-
ly), an abstract representation (no contained '.' in representationPart
string) is equivalent to a resourceSpecificString for an information re-
source. Distinction between a resourceSpecificString and representation part
is defined by naming authority in the resourceType definition.

safeString = safeBoundChar [ *(unreserved / pct-encoded) ] safeBoundChar
safeBoundChar = 1( ALPHA / DIGIT / "_" / "~" )
; first or last character in a safe string is a letter (case insensitive), digit,
or one of only 2 special characters. This restriction is to avoid file-
system name problems across various operating systems.

safeString_np = safeBoundChar
[ *( ALPHA / DIGIT / "-" / "_" / "~" / pct-encoded) ] safeBoundChar ; a
safeString with no contained '.' characters

unreserved = ALPHA / DIGIT / "-" / "." / "_" / "~"
pct-encoded = "%" HEXDIG HEXDIG ; e.g. %20 for space, from RFC3986

```

Interpretation of a USGIN URI

In order to meet the requirement that USGIN URIs are transparent, identifiable, self-describing outlined above, and following the conclusions in Mendelsohn and Williams (2006-12-04), the USGIN URI syntax is designed be interpretable by human users. This section outlines the semantics that are intended for various parts of URIs minted using this scheme.

One of the objectives of USGIN URIs is to define a syntax that makes clear the distinction between an identifier for a non-information resource, information resource and specific representation resource to address the issues raised by Booth (2003-01-28). The basic rules are that (1) a terminal '/' denotes that the identified resource is a

non-information resource, otherwise it is an information resource, and (2) that absence of a terminal '/', and the presence of a '.' in the last segment of the URI denotes that the identified resource is a representation (which is an information resource). If there is no terminal '/' and no '.' in the last segment, then the identified resource is an abstract information resource, like 'USGS home page' or 'current National Weather Service Forecast discussion for Tucson'.

Note that this interpretability is intended to assist users understand what the identifier is identifying. If the identifier is for a non-information resource the URI cannot be interpreted to provide information about the nature of the representation that will be returned when the identifier is dereferenced on the web – this follows the tenants of 'web opacity'. Use of '.' and a file extension to identify a particular representation format is common practice; purists will debate its consistency with web architecture, but it is useful.

Summary of URI syntax: "http:" "//" uriHost "/uri-gin/" nameAuthority "/"
resourcePath "/" [resourceSpecificString] [("/" / "/" representationPart)]

URI for a non information resource: "http:" "//" uriHost "/uri-gin/" nameAuthority "/"
resourcePath ["/" resourceSpecificString] "/" . The terminal '/' denotes that the identified resource is a non-information resource.

URI for an information resource: "http:" "//" uriHost "/uri-gin/" nameAuthority "/"
resourcePath "/" safeString_np . Absence of a terminal '/' denotes that the identified resource is an information resource. Absence of a '.' indicates that identifier is not for a specific representation. The resourceType definition should specify dereferencing behavior when specific representation is not identified.

URI for Specific representation of a resource: "http:" "//" uriHost "/uri-gin/" nameAuthority
"/" resourcePath ["/" resourceSpecificString] [("/" / "/" representationPart)]. Absence of a terminal '/', and the presence of a '.' in the last segment of the URI denotes that the identified resource is a representation (which is an information resource).

The following section has more detailed discussion of the significance and usage of the individual parts of the USGIN URI.

uriHost -- The intention of this segment of the URI is to specify a host system on the internet that will dereference the following identifier. This segment identifies a host server via the DNS registry, and not considered part of the USGIN resource identifier for string comparisons of identifier equivalence. The host name may be a under a separate name authority from the USGIN URI; thus the nameAuthority in the URI is the steward of the resource identifier. Authoritative identification of USGIN name authorities must be done using the resources.usgin.org host. The uriHost is a sequence of characters intended for lookup using the Domain Name System (DNS) service name registry, uses the syntax defined in Section 3.5 of RFC1034 (Mockapetris, 1987) and Section 2.1 of RFC1123 (Braden, 1989). Such a name consists of a sequence of domain labels separated by ".", each domain label starting and ending with an alphanumeric character and possibly also containing "-" characters. RFC3986 (Berners-Lee et al., 2005) states that URI producers should limit these names to no more than 255 characters in length.

resourcePart -- The sequence of characters between "uri-gin/" and the last "/" in URI constitute an identifier for a resource that may have several possible associated information resources, which may be specific representations (document instances), or information resource that are abstract documents (sensu Booth, 2003-01-28) that have one or more representations. One of these representations is the default or canonical representation that is the information resource returned by an http GET request using the URI without a terminal representation. Each resource type definition should specify what the default and canonical representations are (they may be the same or different), and specify the dereferencing behavior for the resource type.

nameAuthority A sequence of characters that identifies the naming authority for the identified resource, taken from the register with USGIN URI http://resources.usgin.org/uri-gin/usgin/register/nameAuthority/. This

URI should dereference to access a list of registered name authorities, along with a description of other services that may be accessed for name authorities.

`resourceType` is a token that indicates the resource type, taken from the register at <http://resources.usgin.org/uri-gin/usgin/register/resourceType/>. The `resourcePath` is hierarchical to allow representation of resource hierarchy when that is useful. Resource type definitions should specify the known valid representations, and dereferencing behavior for URIs for the resource type.

`resourceSpecificString` A string that may have syntax specially scoped for a particular resource type. Definition of `resourceSpecificString` syntax should be in the resource type definition. The string should include version information if appropriate. Recommended practice for a term in a vocabulary is to include the `resourceSpecificString` for the vocabulary, followed by “.”, and then a string identifying the term in the scope of that vocabulary (see concept example, below).

`representationPart` is intended to identify a representation of the preceding resource part. If a “.safestring” segment is present, the `representationPart` identifies a specific representation. If no “.” is present, the `representationPart` identifies an abstract document. Dereferencing behavior for such an identifier is defined in the associated resource type definition. Note that an abstract document is an information resource, so in this case the `representationPart` is semantically equivalent to a `resourceSpecificString`.

The syntax of this identifier is intended to be a valid file name that will work on any current operating system. See File name restrictions for a summary of the rationale for rules governing file name syntax.

Given a USGIN URI, it is possible to guess identifiers for related resources that may be dereferenced to learn more about the identified resource. For example, given an identifier <http://resources.usgin.org/uri-gin/azgs/document/map/DGM37-HuachucaMountainN/> for the concept of the geologic map of a particular footprint area the following other URIs can be deduced:

<http://resources.usgin.org/> -- identifies a registry host. Dereferencing should provide information about the stewardship of the computer system and registry implementation.

<http://resources.usgin.org/uri-gin/> -- identifier for the URI scheme. Dereference for document describing the URI scheme

<http://resources.usgin.org/uri-gin/azgs/> -- identifier for a naming authority. Dereference to learn about the identity of the naming authority. The canonical resource representation should include information about the hostURI instances known to the naming authority. Discrepancy between the naming authority and the hostURI should be reason to suspect the reliability of the dereferencing service.

<http://resources.usgin.org/uri-gin/azgs/doc/> -- identifier for a resource type. Dereference to learn about the resource, and services that may be accessible to provide additional functionality for this resource type.

<http://resources.usgin.org/uri-gin/azgs/doc/map/> -- identifier for a resource type that is a subtype of the document/ resource. Dereference to learn about the resource, and services that may be accessible to provide additional functionality for this resource type.

Identifier equivalence

String comparisons of USGIN URIs to determine equivalence are based on the URI scheme “/” identifier part of the URI. Thus the following URI pairs are equivalent. “<http://resources.usgin.org/uri-gin/azgs/person/steveRichard/cv/cv20100110.doc>” and “<http://geon.org:88/uri-gin/azgs/person/steveRichard/cv/cv20100110.doc>” identify the same representation of the same resource.

Fragments

USGIN URIs do not use document fragments. The URIs are intended to represent an individual resource. In general, a resource that is part of another resource is a different kind of resource, and will have a distinct re-

sourceType path. Binding of parts to the whole should be available in the representations for the whole and part resources. For example a vocabulary term is a ‘classifier’ resource that is part of a ‘conceptScheme’. The conceptScheme resource should include a listing of all classifiers that are part of that collection, and each contained classifier should contain a reference to the ‘owning’ conceptScheme. Recommended approach for binding between a part and some containing resource (e.g. a term in a vocabulary) in the URI is to use a ‘.’ hierarchy in the resourceSpecificString (see concept example below). Individual authorities may choose to implement dereferencing of fragments against URIs for container resources, but the URL using the URI fragment syntax (‘#’) should not be defined as the normative URI for the resource.

Versions

No formal syntax is mandated for encoding version information in this URI scheme. Users are strongly encouraged to develop a system for including version information in identifiers for resources that are versioned. This system should distinguish between different versions, as well as identify the ‘current’ or ‘normative’ version (if these are different, and need to be identified...).

Canonical forms

A canonical representation should be specified as part of the definition of a resource type, and all URI that identify instances of that resource type are expected to dereference to the appropriate canonical form in the absence of an http accept header specifying a different representation type. Canonical forms are a function of the resource type. For best interoperability, a community of practice should have a public registry for resource types that includes a binding of the resource type with definition of the canonical information resource for that resource type, and the expected de-referencing behavior when an identifier is dereferenced.

Note that the canonical form is typically a particular representation of a document, like a formatted text document intended for people, or using some kind of formal syntax indented for machine consumption. This document has an identifier as a ‘document’ resource; association with another resource as a canonical form is a role. Thus the document has two identifiers, one for the document itself, and one for the document in its role as the canonical representation of some other resource. The binding between these two identifiers must be maintained independently in the resource registry.

Functional Resources

Functional resources are those that perform operations activated by user input via the web, or based on the content of the URL used to get the resource. An operation is a non-information concept—it can be invoked electronically, and the result can be transmitted electronically, but the actual activity can not. The concept of a representation for a functional resource is tricky. Dereferencing a URI for a functional resource should provide access to the functionality of the resource—thus may present a web interface for the resource, or enable download of a file containing code that will provide the functionality on the client machine. For practical purposes, USGIN proposes a non-information abstract view of a functional resource like “spreadsheetApplication/MicrosoftExcel/” (to identify the Microsoft Excel Application), an information view that would be a particular software implementation of the functionality: “spreadsheetApplication/MicrosoftExcel/-winExcel11_8316_8221”, and a particular representation would be a file format: “spreadsheetApplication/MicrosoftExcel/winExcel11_8316_8221.msi”. See Examples section.

Meta-resource identifiers and registry structure

This section pertains to the implementation of a registry for USGIN URIs to allow dereferencing using web server and file-system paths. The defined URI syntax requires a URIScheme, nameAuthority, and resourceType in any URI. Registration of resource types is in the scope of particular nameAuthorities, thus each name authority must have a registry/resourceType resource that is a vocabulary of resource categories defined by that authority.

URIScheme (e.g. uri-gin) are defined outside the scope of an individual repository, but the USGIN name authority will maintain a registry of known URISchemes based on the USGIN URI pattern at registry/uriScheme. The resources registered in this listing will include identification of the authoritative uriHost for that scheme.

nameAuthorities are registered at the level of the URIScheme itself, thus a special URI is used to identify top level nameAuthorities: `http://resources.usgin.org/uri-gin/authority/%name%`. This construction is used mostly for branding purposes such that top-level authorities are not presented as subsidiary to the USGIN organization. Any name authority registered at this level may defined sub authorities within the purview of that authority using URIs like “`http://resources.usgin.org/uri-gin/azgs/authority/azgs.mapping`”

Comparison with other schemes

UK public sector

Mandates use of http URIs.

```
"http:" "/" uriHost "/" uriType "/"
  resourcePath "/" [ resourceSpecificString ] [( "/" / "/" representationPart )]
uriType = ("id" / "doc" / "def" / "set")
```

Elements not defined here are reused from the *USGIN URI syntax* section, above. This scheme implies the name authority by the use of data.gov.uk in the uriHost element. The URI types are defined somewhat fuzzily in Davidson (2009-10):

id -- the URI identifies a non-information resource

doc -- the URI identifies a document (a particular kind of information resource) or a representation (which is always some kind of document)

def -- URI identifies a concept-- a non information resource that may be used to categorize other resources

set -- URI identifies a registry, or list of identifiers scoped to some unifying concept (e.g. schools, U. of Az master's theses)

The system does not appear to account for identifying events or functional entities. The resource path is modeled as a sequence of ‘Concept/Reference’ pairs, in which the concept specifies a kind of thing, and reference specifies an instance of the concept.

Examples

This section is a collection of example URIs with some notes on what they identify and usage. These have mostly been invented for didactic and experimental purposes, and are NOT NORMATIVE. In the future the URIs in this section will be real URIs that will dereference according to the provisions of the scheme outlined here. To emphasize that the uriHost name is not part of the actual identifier, the token ‘`http://%uriHost%/`’ is used instead of an actual registered domain name, except for the normative name authority registry for the USGIN URI scheme, which resides at `http://resources.usgin.org`.

Special URIs

Three special URI forms are defined for use with this URI scheme. These are identifiers for a URI host, a URI profile, and the authorities recognized for use with the URI profile. These ‘metalevel’ URIs are necessary as a foundation for trust that identifiers are properly dereferenced.

URI host

`http://resources.usgin.org/` ; special URI that identifies the URI host. This URI should dereference to an HTML page that identifies the authority maintaining the host, and the URI profiles that are known to this host.

URI profile

`http://resources.usgin.org/uri-gin/` ; special URI for the URI profile ('uri-gin' in this example). Assuming this profile is one of the URI profiles known to the host, this URI should dereference to an HTML page explaining the purpose and scope of the URI profile, the authority responsible for the profile, a list of URI hosts trusted to dereference URIs in this profile, and a link to a normative document specifying the syntax and interpretation of URIs conforming to the profile.

Naming authority

`http://resources.usgin.org/uri-gin/usgin/` ; special URI for top level name authority usgin under the uri-gin URI scheme. Assuming the URI profile ('uri-gin') is one of the profiles known to the host, this

`http://resources.usgin.org/uri-gin/azgs/` ; special URI for top level name authority azgs under the uri-gin URI scheme. This authority is recognized to define subauthorities (see below)

`http://%uriHost%/uri-gin/azgs/authority/azgs.mapping/` ; URI for the azgs.mapping subauthority under the azgs name authority.

`http://%uriHost%/uri-gin/azgs.mapping/document/map/DGM37-HuachucaMountainN/` ; The mapping group is a sub authority under the azgs authority, responsible for assigning URI's to maps. (see Map examples for discussion of map URI).

Resource URIs using the uri-gin profile

In these URIs, the string '%uriHost%' is used instead of a particular domain name to make it clear that the actual identifier is the part after the host identifier string.

Person

`http://%uriHost%/uri-gin/azgs/person/StephenRichard/` ; identifies non-information resource that is physical entity.

Vocabulary (codelist)

`http://%uriHost%/uri-gin/cgi/conceptScheme/simpleLithology200811/` ; identifies a particular version of a vocabulary. Dereferencing behavior is defined in resource type definition for urn-gin/cgi/conceptScheme

`http://%uriHost%/uri-gin/cgi/conceptScheme/simpleLithology200811/SimpleLithology200811.skos.rdf` ; skos encoded representation of the vocabulary.

`http://%uriHost%/uri-gin/cgi/conceptScheme/simpleLithology200811/SimpleLithology200811.xls` ; Microsoft Excel encoded representation of the vocabulary.

Term, concept, classifier

`http://%uriHost%/uri-gin/cgi/classifier/simpleLithology200811.granite/` ; identifier for non-information resource. Dereferencing behavior is defined in resource type definition for urn-gin/cgi/classifier

<http://%uriHost%/uri-gin/cgi/classifier/simpleLithology200811.granite/image> ; image representation of simpleLithology200811.granite. Dereferencing behavior is defined in resource type definition for urn-gin/cgi/classifier

Organization

<http://%uriHost%/uri-gin/azgs/organization/arizonaGeologicalSurvey/>

<http://%uriHost%/uri-gin/azgs/organization/USGeologicalSurvey.WRD/> ; This URI represents an organization within an organization.

<http://%uriHost%/uri-gin/azgs/organization/arizonaGeologicalSurvey/azgsIcon> ; this URI identifies an 'abstract' document that is the canonical representation of the icon for AZGS.

http://%uriHost%/uri-gin/azgs/organization/arizonaGeologicalSurvey/azgsIcon300_300.tif

<http://%uriHost%/uri-gin/azgs/organization/arizonaGeologicalSurvey/azgsDefinition.pdf>

<http://%uriHost%/uri-gin/azgs/organization/arizonaGeologicalSurvey/AZGShomepage.html>

Map

<http://%uriHost%/uri-gin/azgs/doc/map/DGM37-HuachucaMountainN/> ; This URI represents the concept of the geologic map of a particular footprint area. This map may change through time, and be represented in various ways. The identified resource is an image. There is a conceptual relationship between this image and some dataset concept on which the image is based.

<http://%uriHost%/uri-gin/azgs/doc/map/DGM37-HuachucaMountainNv1.1/> ; This URI represents the concept of a particular version of a geologic map of a particular footprint area. This map may be represented in various ways (formats).

<http://%uriHost%/uri-gin/azgs/doc/map/DGM37-HuachucaMountainNv1.1/mapImageFile>; A tiff file containing the canonical image of a particular version of the geologic map.

<http://%uriHost%/uri-gin/azgs/doc/map/DGM37-HuachucaMountainNv1.1/mapImageFile.pdf>; A file containing the image of a particular version of the geologic map in a particular format (pdf).

<http://%uriHost%/uri-gin/azgs/doc/map/DGM37-HuachucaMountainNv1.1/hardCopy>; identifies the canonical hard copy version of the map--e.g. printed in color at a particular scale.

<http://%uriHost%/uri-gin/azgs/doc/map/DGM37-HuachucaMountainNv1.1/hardCopy.bw100K>; identifies the hard copy version of the map--printed in black and white at a different scale.

Dataset

<http://%uriHost%/uri-gin/azgs/dataset/geologyMap/HuachucaMountainNv1.1/>; identifies a collection of data describing the geology of a quadrangle.

<http://%uriHost%/uri-gin/azgs/dataset/geologyMap/HuachucaMountainNv1.1/NCGMP09>; identifies a collection of data describing the geology of a quadrangle, in a particular database schema.

<http://%uriHost%/uri-gin/azgs/dataset/geologyMap/HuachucaMountainNv1.1/NCGMP09.shp.zip>; identifies a collection of data describing the geology of a quadrangle, represented in a particular database schema, encoded in shape files and dBase tables, bundled in a zip archive

`http://%uriHost%/uri-gin/azgs/dataset/geologyMap/geoPoly/HuachucaMountainNv1.1/`; identifies a collection of polygon data describing geologic unit outcrops in a quadrangle.

Feature

`http://%uriHost%/uri-gin/azgs/feature/mappedFeature/geoPoly/HuachucaMountainNv1.1-346602/`; identifies a particular polygon in a dataset. Data set membership is scoped in the resource-specific string part of the URI, and would be expected to match the resource specific string for a dataset URI that contains the poly. This

`http://%uriHost%/uri-gin/azgs/feature/geologicUnit/EscabrosaFormation/`; a particular geologic unit.

Document

`http://%uriHost%/uri-gin/usgin/doc/USGIN_ISO_metadata``http://%uriHost%/usgin1_1/` ; this URI identifies an 'abstract' document named `USGIN_ISO_metadata1_1`

`http://%uriHost%/uri-gin/usgin/doc/usginUSGIN_ISO_metadata1_1.doc` ; this URI identifies the '.doc' representation of the abstract document named `USGIN_ISO_metadata1_1`

`http://%uriHost%/uri-gin/usgin/doc/usginUSGIN_ISO_metadata1_1.pdf` ; this URI identifies the '.pdf' representation of the abstract document named `USGIN_ISO_metadata1_1`

Web application

`http://%uriHost%/uri-gin/azgs/webApplication/searchBibliography/` ;URI for database search web application. Dereference will access whatever is the current resource for conducting the search

`http://%uriHost%/uri-gin/azgs/webApplication/searchBibliography/azgeobib0101.html` ;URI for a particular bibliography search web application; this resource bundles a collection of operations, and the dataset that is available to invoke operations on. Technically, the representation is whatever php, javascript, html, xml etc. code is presented to the client that dereferences this identifier. The content that the resource accesses is a different resource—a bibliography dataset in this case.

Application

`http://%uriHost%/uri-gin/azgs/application/stationDataEntry20100225/` ; URI for functionality included in station data entry application. Since this functionality will change over time, the resource specific string should generally include a version identification part.

`http://%uriHost%/uri-gin/azgs/application/stationDataEntry20100225/windowsInstaller` ; URI for current canonical windows installer application for the station data entry application.

`http://%uriHost%/uri-gin/azgs/application/stationDataEntry20100225/windowsInstaller.msi` ; URI for current canonical windows msi installer (as opposed to exe or jar or war or zip) application for the station data entry application.

Service

`http://%uriHost%/uri-gin/azgs/service/WMS/azGeology/` ; URI for functionality of Arizona Geology web service; this resource bundles a collection of operations, and

the dataset(s) available to invoke operations on. Recommendation is that canonical representation obtained by dereferencing a service URI should produce a service description document like an OGC getCapabilities or WSDL document.

Mapping to CGI URN

Examples of USGIN URIs generated from CGI URNs in the CGI registry (start at <https://www.seegrid.csiro.au/twiki/bin/view/CGIModel/CGIRegisterRegister>)

urn:cgi:register:CGI:resourceClass -- <http://%uriHost%/uri-gin/cgi/register/resourceClass/> or <http://cgi.iugs.org/uri-gin/cgi/register/resourceClass> or http://cgi.iugs.org/uri_gin/cgi/register/resourceClass if CGI wants to define different conventions for use of the URI scheme

urn:cgi:register:CGI:classifierScheme -- <http://%uriHost%/uri-gin/cgi/register/classifierScheme/>

urn:cgi:resourceClass:featureType-- <http://%uriHost%/uri-gin/cgi/resourceClass/featureType/>

urn:cgi:resourceClass:serviceType -- <http://%uriHost%/uri-gin/cgi/resourceClass/serviceType/>

urn:cgi:resourceClass:classifier -- <http://%uriHost%/uri-gin/cgi/resourceClass/classifier/>

urn:cgi:propertyType:CGI:GeoSciML:2.0:GeologicEvent:eventAge -- <http://%uriHost%/uri-gin/cgi/property/geologicEvent.eventAge/>

urn:cgi:classifierScheme:CGI:ConsolidationDegree:200811-- <http://%uriHost%/uri-gin/cgi/classifierScheme/consolidationDegree200811/>

urn:cgi:classifier:CGI:ConsolidationDegree:200811:unconsolidated -- <http://%uriHost%/uri-gin/cgi/classifier/consolidationDegree200811.unconsolidated/>

urn:cgi:party:CGI:BRGM -- <http://%uriHost%/uri-gin/cgi/party/organization/brgm/>

urn:cgi:party:CGI:AZGS -- <http://%uriHost%/uri-gin/cgi/party/organization/azgs/>

urn:cgi:registerItem:CGI:authority:BGS -- <http://%uriHost%/uri-gin/cgi/authority/bgs/>

Resource Types

Table 1. Register of resource types recognized by USGIN system. Note this compilation should be considered an example, NOT NORMATIVE.

Resource Type Label	URI type token	Broader Resource Type	Source	Canonical and default representation	Definition
Collection	collection		DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/	Canonical form is xml file containing metadata record describing the collection, including information about how to access collection items. Default form is web page that includes a definition of the purpose of the collection, the agent who maintains the collection, instructions for accessing the collection and a table with URI's for all items in the collection,	An aggregation of resources. A collection is described as a group; its parts may also be separately described. (from http://www.ukoln.ac.uk/metadata/dcmi/collection-application-profile/): The term "collection" can be applied to any aggregation of physical or digital items. Those items may be of any type, so examples might include aggregations of natural objects, created objects, "born-digital" items, digital surrogates of physical items, and the catalogs of such collections (as aggregations of metadata records). The criteria for aggregation may vary: e.g. by location, by type or form of the items, by provenance of the items, by source or ownership, and so on. Collections may contain any number of items and may have varying levels of permanence. A "collection-level description" provides a description of the collection as a unit: the resource described by a collection-level description is the collection, rather than the individual items within that collection. Collection-level descriptions are referred to in Michael Heaney's <i>An Analytical Model of Collections and their Catalogues</i> as "unitary finding-aids".
Vocabulary	vocabulary	Collection		see Collection	A collection of concepts, each with one or more labels, source citation, a definition, and links to associated resources like owl definition, photos, etc. Should ConceptScheme also be a valid resource?

Dataset	dataset	Collection	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/	see Collection	A collection of data items in which individual data items are identified and accessible. DCMI definition is "Data encoded in a defined structure." with additional comment "Examples include lists, tables, and databases. A dataset may be useful for direct machine processing." The container may be a stand-alone digital file (mdb, spreadsheet, table in a Word document), a web service, or an enterprise database. Metadata for the collection is a different type than metadata for individual items in the collection. Criteria for what unifies the collection are variable (topic, area, author...). Synonym: structured data collection. This resource type represents the intellectual artifact - the information content and organization; the dataset may have more than one manifestation (format) -- as a list, a table, databases, using different software implementations.
Register	register	Dataset			A collection of data items each of which is a normative association between a resource and an identifier. May include association with other resources related to the registered resource. Typically registered items are non-information resources. Each register should specify the canonical representation used for items in the register. Canonical representation of register is
Catalog	catalog	Dataset	USGIN		A collection of data items that index resources, as in metadata records; a metadata registry. The resource represents the information content and organization. Catalogs are accessed using other resources, like an interactiveResource or Service, and may have different formats.
Physical artifact collection	collection/physicalArtifact	Collection	USGIN		A collection of identifiable physical objects, unified based on some criteria. Criteria for defining a collection may be who collected, where curated, why collected, kind of material...

Document	doc		USGIN		A packaged body of intellectual work; has an author, title, some status with respect to Review/authority/quality. USGS peer reviewed would be a 'status property'. Have to account for gray literature, unpublished documents, etc. A document may have a variety of physical manifestations (pdf file, hardbound book, tiff scan, Word processor document...), and versions may exist as the document is traced through some publication process. May be map, vector graphics, text. Sound, moving images are included as document types.
Image	doc/image	Document	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A visual representation other than text. Comment: Examples include images and photographs of physical objects, paintings, prints, drawings, other images and graphics, animations and moving pictures, film, diagrams, maps, musical notation. Note that Image may include both electronic and physical representations.
StillImage	doc/image/stillimage	Image	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A static visual representation. Comment: Examples include paintings, drawings, graphic designs, plans and maps. Recommended best practice is to assign the type Text to images of textual materials if the intent of the image is to capture the textual content as opposed to the appearance of the medium containing the text. Instances of the type Still Image must also be describable as instances of the broader type Image. Subtype of Image.
Human-generated image	doc/image/drawing	StillImage	USGIN		Image produced by human drawing or painting, using any media. May be entirely product of human imagination, human perception of the world, or a human-modified photographic image.
Photograph	doc/image/photograph	StillImage	USGIN		Image produced by optical device with chemical or electronic image capture; represents things in the field of view directly as captured by the device. Photographs may be modified by human processing; there is a continuum between photographs and human-generated image. Distinction between the two is largely based on intention

Remote sensing Earth image	doc/image/rsimage	StillImage	USGIN		Image of earth surface acquired by an air born or earth-orbiting sensor. May be georeferenced such that location in the image directly corresponds to location on the earth.
Map	doc/map	StillImage	USGIN		Human-generated depiction of some part of the earth using a mathematical system of correspondence between geometry in the image and location on the earth.
Moving image	doc/movingImage	Document	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A series of visual representations imparting an impression of motion when shown in succession. Comment: Examples include animations, movies, television programs, videos, zoetropes, or visual output from a simulation. Instances of the type Moving Image must also be describable as instances of the broader type Image. Subtype of Image. Commonly include sound
Sound	doc/sound	Document	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A resource primarily intended to be heard. Comment: Examples include a music playback file format, an audio compact disc, and recorded speech or sounds.
Text	doc/text	Document	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A resource consisting primarily of words for reading. Comment: Examples include books, letters, dissertations, poems, newspapers, articles, archives of mailing lists. Note that facsimiles or images of texts are still of the genre Text.
Hypertext document collection	doc/hyptertext	Text	USGIN		A collection of files that contains http hyperlinks between them. Links to documents or other resources outside of the collection are possible. The criteria for determining membership in the collection are somewhat arbitrary, but in general the 'site' should contain related documents authored and managed by the same agent.
Event	event		DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A non-persistent, time-based occurrence. Metadata for an event provides descriptive information that is the basis for discovery of the purpose, location, duration, and responsible agents associated with an event. Examples include an exhibition, webcast, conference, workshop, open day, performance, battle, trial, wedding, tea party, and conflagration.

Project	event/project	Event	USGIN		Project represents a funded activity that has some purpose; projects have associated extents, which represent the area of interest for the project. This extent serves as a mechanism to filter descriptions and concepts in the information system for those that may be related to the project based on spatial relationships. Projects in a large organization will likely have hierarchical (part-whole) relationships.
Model	model		USGIN		Algorithm, workflow; an abstract representation of a collection of related processes, objects and relationships. A model resource may be related to various kinds of document that portray the model, or to software that implements the model, or with datasets as input or output. Not clear that there is a compelling use case for cataloging models separately from the software or documents that are manifestations of the model.
Physical object	physicalObject				
Physical artifact	physicalObject/artifact	Physical object	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		General category for physical resources that are indexed by metadata records; also root of an artifact type hierarchy. An identifiable physical object. Identification is always a function of some human intention, thus differentiating an artifact from other 'natural' things. Note that digital representations of, or surrogates for, these objects should use Image, Text or one of the other types.
Party					
Person	party/person	Party			
Organization	party/organization	Party			
Role-based party	party/organization/role	Organization			

Service	service		DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A system that provides one or more functions via a network interface designed for machine interaction. An implementation of an interface to some sort of digital resource, using either a 'pull' model in which client requests some content from the service, and receives that content in a single 'response' package, or a 'push' model in which client establishes connection and monitors for change events (update, new data...) from service. Difficult to draw line on when a service provides 'files' and when it provides 'data', because responses are always in a form that could be considered a file. Also includes interfaces to digital resources that provide a continuous (with some sampling interval?) feed of some sort of data.
Map service	mapService	Service			a service that presents operations to provide map images. A service instance is a binding between a collection of 1 to Many operations, 0 to many datasets, and 0 to many portrayals.
Software	software		USGIN		A computer program in source or compiled form. Comment: Examples include a C source file, MS-Windows .exe executable, or Perl script.
Stand-Alone-Application	application	Software	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		Identifiable stand alone software application. Identity of resource is based on function performed, input and output requirements, and authorship. The same application may be packaged in different file formats to run in different software environments; thus an application will have one or more associated digital files. For the purposes of this catalog scheme, stand alone applications are software that can be packaged in a single file that can be transferred between machines, unpackaged and compiled or installed on a computer meeting specified hardware and software environment conditions, to execute the described function on that computer, independent of any network connection.

Interactive web resource	webApplication	Software	DCMI resource Types http://dublincore.org/documents/dcmi-type-vocabulary/		A resource requiring interaction from the user to be understood, executed, or experienced. Comment: Examples include forms on Web pages, applets, multimedia learning objects, chat services, or virtual reality environments. Interactive resources are software driven. From the point of view of the catalog, they are accessed by a URL to a web site that is the interface for operating the application. The application operates by interaction with one or more human participants. The application requires network connection to operate, is accessible via the internet, and requires human interaction.
Structured digital data item	dataItem		USGIN		An individually identifiable item in a structured digital data collection. Characterized by a schema, and some particular values. In ISO11179 terms, this is an instance of a data element. Tagging, commenting, reviewing, rating community interaction with catalog will probably require metadata records about particular data items in cataloged datasets (including metadata items in catalogs.)
Feature	feature	dataItem			very tricky here--is a feature a representation artifact or a thing in the world
Mapped feature	mappedFeature	feature			
Data record	dataRecord	dataItem			
Sampling point, site, station	samplingPoint	Structured digital data item	From ScienceBase item types, SMR redux		A resource that is a location-based container/base for observation data. Should this be generalized to OGC O&M samplingFrame to include other sampling geometry (borehole, image footprint)... Analogous in function to a keyword, but carries metadata on who located, when, why, how...
Portrayal	portrayal				A binding between some dataset and a collection of portrayal rules. A WMS layer, an ESRI layer file, an SLD with linked data are possible representations.

Table 2. Register of name authorities recognized by USGIN system (version 0). Normative representation of this register is SKOS rdf document with URI <http://resources.usgin.org/uri/usgin/register/authority/authoritySKOS00.rdf>. HTML representation of current version is <http://resources.usgin.org/uri/usgin/register/authority/authority.html> (not implemented yet...)

URI token	Broader Resource Type	Source	Definition
algs	Organization	AASG geothermal data project	Alabama Geological Survey
azgs	Organization	AASG geothermal data project	Arizona Geological Survey
cadoggr	Organization	AASG geothermal data project	California Department of Oil, Gas, and Geothermal Resources
cgi	IUGS	USGIN	International Union of Geological Sciences Commission for the Management and Application of Geoscience Information
cgs	Organization	AASG geothermal data project	Connecticut Geological Survey
cogs	Organization	AASG geothermal data project	Colorado Geological Survey
flgs	Organization	AASG geothermal data project	Florida Geological Survey
isgs	Organization	AASG geothermal data project	Illinois Geological Survey
kgs	Organization	AASG geothermal data project	Kentucky Geological Survey
lgs	Organization	AASG geothermal data project	Louisiana Geological Survey
mags	Organization	AASG geothermal data project	Massachusetts Geological Survey
mtgs	Organization	AASG geothermal data project	Montana Geological Survey
nbgm	Organization	AASG geothermal data project	Nevada Bureau of Mines and Geology
ngmdb	USGS	USGIN	National Geologic Map Database Project
ogs	Organization	AASG geothermal data project	Oklahoma Geological Survey
tbeg	Organization	AASG geothermal data project	Texas Bureau of Economic Geology
usgin	Project	USGIN	U. S. Geoscience Information Network

URI token	Broader Resource Type	Source	Definition
usgs	Organization	USGIN	USGS
utgs	Organization	AASG geothermal data project	Utah Geological Survey
wvgs	Organization	AASG geothermal data project	West Virginia Geological Survey
wygs	Organization	AASG geothermal data project	Wyoming Geological Survey

DRAFT

References

- Booth, David, 2003-01-28, Four uses of a URL: Name, Concept, Web Location and Document instance, Revision 1.29 2003-01-28: accessed at http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm, 2010-02-15.
- Braden, R., 1989-10, Requirements for Internet Hosts - Application and Support: The Internet Society Network Working Group Standards Track, STD 3, RFC 1123.
- Berners-Lee, T., Fielding, R., Masinter, L., 1998-08, Uniform Resource Identifiers (URI): Generic Syntax: The Internet Society Network Working Group Standards Track, RFC 2396 (superseded by RFC-3986), accessed at <http://tools.ietf.org/html/rfc2396> 2010-02-19.
- Berners-Lee, T., Fielding, R., Masinter, L., 2005-01, Uniform Resource Identifiers (URI): Generic Syntax: The Internet Society Network Working Group Standards Track, STD 66, RFC 3986 (obsoletes RFC-2396), 53 p., accessed at <http://tools.ietf.org/html/rfc3986> 2010-02-19.
- Crocker, D., and Overell, P., 2008-01, Augmented BNF for syntax Specifications: ABNF: The Internet Society Network Working Group Standards Track STD 68, RFC 5234 (obsoletes RFC 2234, RFC4234), 16 p., accessed at <http://tools.ietf.org/html/rfc5234> 2010-02-19.
- Davidson, Paul, Editor, 2009-10, Designing URI Sets for the UK Public Sector, Interim paper, v. 1.0: Chief Technology Officer Council, accessed at http://www.cabinetoffice.gov.uk/media/301253/puublic_sector_uri.pdf 2010-03-10.
- Fielding, R., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., 1999-06, Hypertext Transfer Protocol -- HTTP/1.1: The Internet Society Network Working Group Standards Track RFC2616, accessed at <http://tools.ietf.org/html/rfc2616> 2010-02-19.
- Lewis, Rhys, editor, 2007-05-31, Dereferencing HTTP URIs: W3C Draft Tag Finding, 10 p., accessed at <http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14.html> 2010-02-19 (note current version dated 2007-10-04 strikes all content from this document).
- Mendelsohn, Noah, and Williams, Stuart, 2007-01-02, The use of Metadata in URIs: W3C TAG Finding, accessed at <http://www.w3.org/2001/tag/doc/metaDataInURI-31-20070102.html>.
- Mockapetris, P., 1987-11, Domain names - concepts and facilities: The Internet Society Network Working Group STD 13, RFC 1034.
- Thompson, Henry S., and Rees, Jonathan, 2009-06-22, Dirk and Nadia design a naming scheme or Web naming schemes good practices: W3C Consortium, accessed at <http://www.w3.org/2001/tag/doc/justSayHTTP> 2010-02-19.

Appendix 1. Collected ABNF for URI, RFC-3986

Quoted from RFC-3986 (<http://tools.ietf.org/html/rfc3986>)

```
URI = scheme ":" hier-part [ "?" query ] [ "#" fragment ]
```

```
hier-part = "://" authority path-abempty
```

```
; path begins with "/" or is empty
```

```
  / path-absolute           ; begins with "/" but not "///"
```

```
  / path-rootless          ; begins with 1*pchar, e.g. urn:cgi:classifier
```

```

    / path-empty          ; zero characters e.g. foo:?fred

URI-reference = URI / relative-ref

absolute-URI  = scheme ":" hier-part [ "?" query ]

relative-ref  = relative-part [ "?" query ] [ "#" fragment ]

relative-part = "//" authority path-abempty
; path begins with "/" or is empty
    / path-absolute      ; begins with "/" but not "//"
    / path-noscheme      ; begins with a non-colon segment
    / path-empty         ; zero characters
; the relative-part allows path-noscheme but not path-rootless, hier-part al-
; lows path-rootless but not path-noscheme.

scheme = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )

authority = [ userinfo "@" ] host [ ":" port ]
userinfo = *( unreserved / pct-encoded / sub-delims / ":" )
host = IP-literal / IPv4address / reg-name
port = *DIGIT

IP-literal = "[" ( IPv6address / IPvFuture ) "]"

IPvFuture = "v" 1*HEXDIG "." 1*( unreserved / sub-delims / ":" )

IPv6address = 6( h16 ":" ) ls32
    / ":@" 5( h16 ":" ) ls32
    / [ h16 ] ":@" 4( h16 ":" ) ls32
    / [ *1( h16 ":" ) h16 ] ":@" 3( h16 ":" ) ls32
    / [ *2( h16 ":" ) h16 ] ":@" 2( h16 ":" ) ls32
    / [ *3( h16 ":" ) h16 ] ":@" h16 ":" ls32
    / [ *4( h16 ":" ) h16 ] ":@" ls32
    / [ *5( h16 ":" ) h16 ] ":@" h16
    / [ *6( h16 ":" ) h16 ] ":@"

h16 = 1*4HEXDIG
ls32 = ( h16 ":" h16 ) / IPv4address
IPv4address = dec-octet "." dec-octet "." dec-octet "." dec-octet
dec-octet = DIGIT ; 0-9
    / %x31-39 DIGIT ; 10-99
    / "1" 2DIGIT ; 100-199
    / "2" %x30-34 DIGIT ; 200-249
    / "25" %x30-35 ; 250-255

reg-name = *( unreserved / pct-encoded / sub-delims )

```

```

path = path-abempty ; begins with "/" or is empty
      / path-absolute ; begins with "/" but not "//"
      / path-noscheme ; begins with a non-colon segment
      / path-rootless ; begins with a segment
      / path-empty ; zero characters

path-abempty = *( "/" segment )
path-absolute = "/" [ segment-nz *( "/" segment ) ]
path-noscheme = segment-nz-nc *( "/" segment )
path-rootless = segment-nz *( "/" segment )
path-empty = 0<pchar>

segment = *pchar ;may be empty
segment-nz = 1*pchar ;non-empty string
segment-nz-nc = 1*( unreserved / pct-encoded / sub-delims / "@" )
                ; non-zero-length segment without any colon ":"

pchar = unreserved / pct-encoded / sub-delims / ":" / "@"
        ; any character except "\", "?", "#", "[", "]"

query = *( pchar / "/" / "?" )
fragment = *( pchar / "/" / "?" )
pct-encoded = "%" HEXDIG HEXDIG
unreserved = ALPHA / DIGIT / "-" / "." / "_" / "~"
reserved = gen-delims / sub-delims
gen-delims = ":" / "/" / "?" / "#" / "[" / "]" / "@"
sub-delims = "!" / "$" / "&" / "'" / "(" / ")"
            / "*" / "+" / "," / ";" / "="

```

Appendix 2. Collected ABNF for URI, RFC-2396

Quoted from RFC-2396 (<http://tools.ietf.org/html/rfc2396>)

```

URI-reference = [ absoluteURI | relativeURI ] [ "#" fragment ]
absoluteURI  = scheme ":" ( hier_part | opaque_part )
relativeURI  = ( net_path | abs_path | rel_path ) [ "?" query ]

hier_part    = ( net_path | abs_path ) [ "?" query ]
opaque_part  = uric_no_slash *uric

uric_no_slash = unreserved | escaped | ";" | "?" | ":" | "@" |
                "&" | "=" | "+" | "$" | ","

net_path     = "://" authority [ abs_path ]
abs_path     = "/" path_segments
rel_path     = rel_segment [ abs_path ]

rel_segment  = 1*( unreserved | escaped |
                  ";" | "@" | "&" | "=" | "+" | "$" | "," )

```

```

scheme      = alpha *( alpha | digit | "+" | "-" | "." )

authority   = server | reg_name

reg_name    = 1*( unreserved | escaped | "$" | "," |
                ";" | ":" | "@" | "&" | "=" | "+" )

server      = [ [ userinfo "@" ] hostport ]
userinfo    = *( unreserved | escaped |
                ";" | ":" | "&" | "=" | "+" | "$" | "," )

hostport    = host [ ":" port ]
host        = hostname | IPv4address
hostname    = *( domainlabel "." ) toplabel [ "." ]
domainlabel = alphanum | alphanum *( alphanum | "-" ) alphanum
toplabel    = alpha | alpha *( alphanum | "-" ) alphanum
IPv4address = 1*digit "." 1*digit "." 1*digit "." 1*digit
port        = *digit

path        = [ abs_path | opaque_part ]
path_segments = segment *( "/" segment )
segment     = *pchar *( ";" param )
param       = *pchar
pchar       = unreserved | escaped |
                ":" | "@" | "&" | "=" | "+" | "$" | ","

query       = *uric

fragment    = *uric
uric        = reserved | unreserved | escaped
reserved    = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" |
                "$" | ","
unreserved  = alphanum | mark
mark        = "-" | "_" | "." | "!" | "~" | "*" | "'" |
                "(" | ")"

escaped     = "%" hex hex
hex         = digit | "A" | "B" | "C" | "D" | "E" | "F" |
                "a" | "b" | "c" | "d" | "e" | "f"

alphanum    = alpha | digit
alpha       = lowalpha | upalpha

lowalpha    = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |
                "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" |
                "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
upalpha     = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |

```

```
digit = "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |  
      "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"  
      "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |  
      "8" | "9"
```

File name restrictions

Compilation of file name restrictions for Mac OS 9 or later, Windows 2000 and later, Unix (versions?).
Sources: <http://www.portfoliofaq.com/pfaq/FAQ00352.htm>

- Disallowed filename characters. “.” “?” “,” “;” (All OSs).
- >64 filename characters including extension. (Windows: ISO9660+Joliet CD or Hybrid CD partition). May need to be honored to allow for archiving resource directories on CD?
- No extension - extensions are mandatory for Windows and the only means for Portfolio to tell file type. (Windows, Mac OS X).
- Filename has >1 period - Portfolio may misinterpret extension. (Windows, Mac OS X).
- Extension may be wrong, i.e. not 3 characters. (Windows, Mac OS X).
- Illegal characters in path to file - same issue as #1 but for path. (All OSs).
- Deprecated characters in path to file - same issue as #2 but for path. (All OSs).
- Filename may not begin with a period. (Windows not allowed, Mac treats as a hidden file)
- Filename may not end in a period. (Windows not allowed - OS 'throws away' the trailing period when naming/reading so incorrect matching vs. Mac name)
- Names conflicting with some of Win OS' old DOS functions (Not allowed in either upper or lowercase and with or without a file extension or as a file extension: COM1 to COM9 inclusive, LPT1 to LPT9 inclusive, CON, PRN, AUX, CLOCK\$ and NUL)
- Case sensitivity. Windows OSs (and IIS web servers) aren't case sensitive. Most other OSs (and web servers) are.
- Filenames ought not to begin with a hyphen (Unix systems may interpret the filename as a flag to a command line call)