Draft: World Wide digilib – Resource Identifier in ECHO

Robert Casties*

Version 0.6 of June 18, 2003

Contents

1 Digital Resource Identifier DRI

The *Digital Resource Identifier* is a worldwide unique identifier for a digital resource. The resource may be an electronic text, single or multiple digital images, an audiovisual media file or other type of electronic resource that is accessible over the Internet.

The identifier provides a stable point of reference for digital resources in the Internet. The identifier is therefore independent from the address, implementation and directory layout of the location of the resource. The identifier is unique and constant and it can be used in other documents to reference the resource without the risk of having a broken reference in the future because the address or filename of the resource has changed.

The identifier supports infrastructure for the "sustainability" of digital resources to guarantee that not only the identifier always points to the same resource but also the

*IT-Group, Max Planck Institute for the history of science

resource stays available in the Internet. The infrastructure supports backup copies and load balancing mechanisms. The implementation and enduring support of the actual servers and digital resources is in itself mostly an organisational and social challenge that cannot be solved by technological measures alone.

1.1 Structure of the DRI

The *Digital Resource Identifier* has the following properties:

- Total address space of 70 bit, partitioned into a million subspaces of 50 bit for 10^{15} or 1125 billion different resources per subspace.
- The identifier contains only (uppercase) letters and digits.
- The identifier is composed of a 4 character *subspace* or *namespace identifier*, a 10 character *resource identifier* and a 1 character checksum, giving a total of 15 characters for the full DRI.

1.2 Character set

The identifier is composed only of letters and digits. Uppercase and lowercase letters are not distinguished. The resulting character set has $26 + 10 = 36$ characters. Four characters with ambiguous shapes that might lead to errors are omitted: “O” (vs. “0”), “I” (vs. “1” or “l”), “L” (vs. “1” or “I”), and “J” (vs “1” or “I”). The resulting set of 32 characters can be used to represent 5 bit of information.

character	value	character	value	character	value	character	value
0	0	A	10	N	20	Y	30
1	1	B	11	P	21	Z	31
2	2	C	12	Q	22		
3	3	D	13	R	23		
4	4	E	14	S	24		
5	5	F	15	T	25		
6	6	G	16	U	26		
7	7	H	17	V	27		
8	8	K	18	W	28		
9	9	M	19	X	29		

Table 1: Character set for identifier

The 50 bit of the chosen address for the resource is divided into ten pieces of 5 bit. The pieces are each encoded into one character according to the character table in table 1. The resulting string of 10 characters is called the *resource address*.

1.3 Namespaces

The total address space of 70 bit is divided into 2^{20} (1048576) subspaces of 50 bit. These subspaces, also called namespaces, can be assigned to institutions that wish to implement their own allocation of resource identifiers for reasons of efficiency and maintenance. All resulting resource identifiers are only valid once they are registered with the central *resource registry*.

Each subspace is identified by a four-character *name space identifier*. The 10 character *resource address* is prefixed with the *name space identifier*, resulting in a 14 character *unique address* for each resource.

Subspaces and their name space identifier are registered by the central resource registry. An institution or project that wishes to implement its own allocation of resource identifiers contacts the resource registry and receives a name space identifier for a currently unused subspace. The subspace is then marked as being used by this institution or project. New resource identifiers in this subspace can only be assigned by the institution or project that owns the subspace.

The central resource registry allocates and registers resource identifiers for institutions, projects and individuals that do not want to maintain their own subspace. Resource identifiers allocated by the central resource registry are in the `ECHO` namespace.

The namespaces `0000`, `TEMP` and `ECHO` are reserved for use with the central resource registry.

1.4 Checksum

A checksum of one character (5 bit) is calculated over the 14 characters (70 bit) of the *unique address*. The checksumming method is similar to the method used for ISBN (International Standard Book Number). The differences are the number system, which is base-32 for the DRI (ISBN: base-10) and the modulus, which is 31 for the DRI (ISBN: 11).

The checksum number is calculated with the formula

$$c = \sum_{i=1..14} i x_i \pmod{31}$$

The resulting checksum number c is converted to a character according to table 1 and appended to the end of the *unique address* giving the full *Digital Resource Identifier*.

The DRI is only valid if the checksum calculated over the unique address part of the identifier (the first 14 characters) matches the checksum value (the last character).

2 Central resource registry

The central resource registry is the keystone in the concept of stable and sustainable digital resource identifiers and references. Resources can be moved and renamed on local servers, duplicated onto other servers and servers can even be shut down (given the resource had been duplicated) without resources getting lost or breaking links or references to the resource.

The resource registry server acts as a switchboard between the user requests for a resource and local servers providing the resource. URLs and other so called “global” references to a resource via its DRI access the resource registry server that dispatches the request to the local server. In this way only the resource registry server’s address has to remain stable.

This places a high burden of availability on the registry server. This challenge can be met on a technical level with standard technology (transparent replication and load balancing) and scaled to higher performance levels when the demand rises. More importantly a durable solution has to be established on the organizational and social level for running the server.

The resource registry maintains the mapping database between the digital resource identifiers and the location of the resources on the local servers. In this way it has a list of all known resource identifiers and ensures that all resource identifiers are unique.

The database on the resource registry server can additionally store a set of minimal meta informations on the resources and provide searches in this metadata. One item of this minimal meta information should be a URL to further information on the resource.

The resource registry server provides a HTTP redirect function for transparent HTTP access to resources and optionally other webservice access (XML-RPC, SOAP).

Special client software for accessing resources can harvest and cache DRI mappings from the central registry for short times to improve performance or offline work.

As mentioned in chapter 1.3 parts of the resource identifier address space can be assigned to institutions or projects to implement their own allocation of resource identifiers. These identifiers are generally valid only after they have been registered with the central resource registry.

The central resource registry remains the only authoritative source of digital resource identifiers and their mapping to local resources.

The resource registry provides interfaces to

- redirect HTTP requests with resource identifiers to local resource servers
- query the mapping of resource identifiers using a webservice interface
- hand out new resource identifiers and acquire the necessary mapping information
- change resource mapping information or resource meta information
- query the database for meta information
- upload sets of externally allocated resource identifiers
- download sets of identifiers or the whole database for caching purposes.

2.1 Handling of digital resource identifiers in HTTP requests

A global HTTP request usually accesses a digital resource via some kind of display tool (for example `digilib`) that is able to render a web representation of the resource. While the resource identifier is embedded in the DRI part of the URL, other aspects of the rendering (for example which tool to use) are embedded in other parts of the URL that may be specific to the display tool. Therefore the registry server has to treat URLs differently depending on the display tool.

The handling of HTTP requests has three steps:

1. Identification of the DRI in the request string.
2. Lookup of additional information on the handling of the request based on the DRI.
3. Redirect of the client to the local resource server.

The first part of the treatment of the URL is the identification of the DRI in the HTTP request string. Three basic ways of handling the DRI are envisaged:

- The DRI can be embedded as part of the URI path[1] (`http://driserver.echo.eu/dri/ECHO00001A2B3CX`),

[1] The first part of the URI path, separated by slashes, that is a valid DRI string.

- it can be provided as a special HTTP GET or POST parameter for a defined environment like `digilib`[2] (`http://driserver.echo.eu/digilib/digilib.jsp?dri=ECHO00001A2B3CX&pn=5`) or
- it can be extracted from the request by a generic pattern matching scheme (this option is computationally most expensive)

Once the DRI is identified more information about the resource can be looked up in the central resource database. From this point on the redirection of the request can be handled differently depending on the record type information in the database.

An extensible set of URL rewrite rules will be implemented by the server. The type of rule to be used is part of the resource record of the DRI in the central resource registry. The following rules should be part of the first implementation of the registry server:

redirect only the host part of the URL is replaced by the local host name from the resource record.

replace the full URL is replaced by the local URL from the resource record.

`digilib` the host part of the URL is replaced by the local host name from the resource record and the remaining part is replaced according to `digilib` rules.

rewrite the host part of the URL is replaced by the local host name from the resource record and the remaining part is replaced according to generic substitution rules with wildcard patterns.

The introduction of other specialized types of rewrite rules can be implemented as extension modules to the resource server.

2.1.1 Redirect and replace type DRI resolution

When a DRI resource record has a resolution type of “redirect”, then only the host part of the URL is replaced in the redirected request by the local host given in the resource record. See table 2.

incoming request	`http://driserver.echo.eu/dri/ECHO00001A2B3CX`
`local_host` record	`penelope.unibe.ch`
redirect request	`http://penelope.unibe.ch/dri/ECHO00001A2B3CX`

Table 2: redirect type DRI resolution

When a DRI resource record has a resolution type of “replace”, then the whole URL is replaced in the redirected request by the local URL given in the resource record. See table 3.

[2]The environment itself should be identified by the first parts of the URI path.

incoming request	`http://driserver.echo.eu/dri/ECHO00001A2B3CX`
`local_url` record	`http://penelope.unibe.ch/docuserver/compago/compare.pl?32`
redirect request	`http://penelope.unibe.ch/docuserver/compago/compare.pl?32`

Table 3: replace type DRI resolution

2.1.2 `digilib` type DRI resolution

When a DRI resource record has a resolution type of "`digilib`", then the host part of the URL is replaced by the local host in the resource record and the remaining part is replaced according to `digilib` parameter format.

In the preferred parameter-style format the DRI is given as the parameter "dri". The local URL for the redirect is constructed by replacing the URI path up to the "?" with the digilib path from the resource record and adding a local filename as parameter "fn". See table 4.

incoming request	`http://driserver.echo.eu/digilib/digilib.jsp?dri=ECHO00001A2B3CX&pn=5`
`local_host` record	`penelope.unibe.ch`
`digilib_path` record	`/docuserver/digitallibrary/digilib.jsp`
`digilib_file` record	`public/Beispiele`
redirect request	`http://penelope.unibe.ch/docuserver/digitallibrary/digilib.jsp?dri=ECHO00001A2B3CX&fn=public/Beispiele&pn=5`

Table 4: digilib type DRI resolution

In the deprecated plus-style format the DRI could be placed the first part of the parameter path, prefixed with "dri:". In the local URL the local pathname is appended to the DRI part.

2.1.3 Rewrite type DRI resolution

When a DRI resource record has a resolution type of "rewrite", then the host part of the URL is replaced by the local host name from the resource record and the remaining part is replaced according to generic substitution rules with wildcard patterns.

2.2 Handling of digital resource identifiers as a web service

The basic function of resolution of a DRI as well as other maintenance functions like the registration of new DRIs or the download of parts or all registered DRI mappings should also be accessible with a web service interface.

Specifications for the web service interface have to be established.

3 Resource metadata

The set of metadata about a resource that is stored on the resource server is called a *resource record*. Since the requirements of access, structure and amount of metadata for different projects can hardly be generalized the resource server stores only a minimal set of fields that is sufficient for the basic functions of access to the resource, sustainability of access, and interoperability. More extensive and project specific metadata sets should be stored and maintained on external servers. The optional resource information field can be used to point to external metadata representations.

3.1 Basic metadata

The amount of metadata is dependent on the type of resource record. Common to all records is the `dri` field for the resource identifier. Redirect-type records require an additional `local_host` field for the host name of the local host. Replace-type records require an `local_url` field for a full URL. Digilib-type records require at least the three fields `local_host`, `digilib_path`, and `digilib_file` and an optional parameter `digilib_pageno`. The basic fields can be found in table 5.

type	field	description
redirect		
	`record_type`	type of record (“redirect”)
	`dri`	DRI
	`local_host`	local host name
replace		
	`record_type`	type of record (“replace”)
	`dri`	DRI
	`local_url`	full local URL
digilib		
	`record_type`	type of record (“digilib”)
	`dri`	DRI
	`local_host`	local digilib server
	`digilib_path`	URI path of the digilib installation
	`digilib_file`	digilib path name (parameter fn)
	`digilib_pageno`	optional page number (parameter pn)

Table 5: Basic metadata fields

The resource server may implement additional fields like owner and group fields for internal management and user access functions.

3.2 Alternate server and backup server

The resource server architecture is designed to fulfill high demands on the performance and sustainability of access to the resources. These demands can be met by a loosely coupled network of local servers duplicating content for backup and the transparent sharing of concurrent access to resources for enhanced performance.

Backup server fields give the names and paths of servers that provide copies of the resource. Requests for the resource are diverted to a backup server when the original server becomes unavailable.

Alternate server fields give the names paths of servers that provide copies of the resource. Requests for a resource are spread among all alternate servers for the same resource according to a load-balancing pattern. The pattern can be a simple round-robin scheme or a more sophisticated scheme based on server performance or the geographical location of client and server.

A resource record can have any number of backup server and alternate server fields. If a resource is required to have at least one backup server is a policy decision of the hosting project that is not enforced by the resource server.

3.3 Additional resource information

The resource server itself carries only minimal metadata on a resource but it provides a basic mechanism to store and access more extensive information on external servers.

Every resource record can have a resource info URL that is stored in the `info-url` field.

field	description
`info-url`	URL to external information

Table 6: External resource information

The external resource information can be accessed in a standardized way on the resource server where the DRI of the resource is part of the URI path: `http://driserver.echo.eu/resinfo/ECHO00001A2B3CX/` Requests to this URL will be redirected to the URL in the `info-url` field in the resource record.