DOI and Internet identifier specifications
Version 1.0

A standard represents an agreement by a community to do things in a specified way to address a common problem. Whilst the DOI community has developed the DOI, it has also ensured conformance with relevant generic external formal standards. This note discusses those relevant in the Internet communities IETF and W3C. There is currently considerable debate here on the issue of generic standards for naming objects. DOI is capable of being used in any specification which may finally be endorsed. Until a clear consensus is reached in the Internet communities on which approach is to be preferred DOI remains agnostic as to formal registration as a generic scheme, but useable and widely implemented for millions of objects.

Generic identifier standards
Persistent and actionable object names are required for coherence in the digital realm.

- “Persistent and actionable object names” thus necessarily require mechanisms for persistence (provided by social infrastructure); actionability (resolution from a name to some service); specification of an object (either through simple referencing or more formal description); and naming syntax (prescriptive rules for assigning identifiers in a standard format and ensuring uniqueness).
- The DOI system uses as its naming syntax the NISO standard Z39.84 “Syntax for the DOI”. DOI uses for its name resolution the Handle System (IETF RFCs 3650, 3651, 3652). DOI uses for its optional object specification a DOI data model including the indecs Data Dictionary and its subset the ISO MPEG 21 Rights Data Dictionary, ISO/IEC 21000-6. (The data dictionary component is designed to maximise semantic interoperability with existing metadata element sets; the data model allows descriptions to be grouped in meaningful ways so that certain types of DOIs all behave the same way in an application). DOI persistence is guaranteed through the IDF social infrastructure which provides rules for registration, formal resilience procedures in the event of any single agency failing, etc.
- DOI conforms to the functional requirements of the two generic approaches for naming first-class objects on the Internet: the Uniform Resource Name (URN) and the Uniform Resource Identifier (URI). URI and URN specifications deal only with syntax and (in part) associated implementation through resolution, not with description or persistence policy. Broadly, the URN approach is favoured by IETF and the URI approach by W3C, though there is considerable ongoing debate about each; some documentation on these specifications is incomplete. Crucially, widespread practical implementations of these specifications as object naming do not exist: both URI and URN are specifications, not in themselves working implementations. The DOI is de facto a practical implementation of URI and URN.
• DOI can also be implemented using current URL (http) specifications. The DOI is also a defined Digital Item Identifier within the ISO MPEG 21 multimedia framework specification.

**URI implementation**

The Uniform Resource Identifier (URI) specification is IETF RFC 2396, URI Generic Syntax, currently under revision as RFC 2396 bis. URIs formally encompass URNs as a sub set. In practice, the URI specification defines (1) an implementation more often called the Uniform Resource Locator, a location on a file server, commonly accessed using the http protocol though other protocols are allowed; (2) a syntax for referencing in XML, through which e.g. ISBNs can be specified as URIs. This provides a single framework which can accommodate any other identifier for referencing, but it is not as such persistent (since persistence is not determined by the specification but by the practical implementation).

Conflating these two causes confusion. Unlike URIs, as currently understood, are demonstrably not persistent; redefining them as URIs doesn’t fix that.

• **URL implementation.** Users may resolve DOIs using the URL syntax through the DOI proxy server (http://dx.doi.org). A DOI of the form doi:10.123/456 would be resolved from the address: "http://dx.doi.org/10.123/456". Any standard browser encountering a DOI in this form will be able to resolve it. The use of the DOI proxy server does not interfere with any http requirements, so DOIs may be used with other http-based mechanisms such as OpenURL, PURL, parameter passing, etc. The DOI proxy server is maintained by the IDF and the DOI community for use by all. However performance and functionality can be improved beyond the level achieved through http by using the optional native resolver protocol (Handle) not requiring the use of http. DOI resolution in native resolver form does not require the use of the DNS (Domain Name System), though does of course when used with the proxy http resolver.

• **URI syntax implementation.** In the URI specification, the network path of the URI is implicitly DNS based; there are no real provisions to include systems that are not DNS based. Original URI specifications, and good design practice, assume the URI to be opaque (that is, it is not assumed that software can parse the body of the URI but that it would simply recognize the name of the scheme and hand it off to some other software that understood the scheme). The current URI specification, however, assumes that the initial URI parser will look into every URI, no matter what the scheme, looking for certain meaningful characters such as dot and slash. This version of the URI proposed in RFC 2396 bis is so restrictive that it is difficult to see what system could make use of it.

• A specification for DOI as a URI exists as an Internet Draft: this document defines the 'doi' Uniform Resource Identifier (URI) scheme for DOIs, which allows a DOI to be referenced by a URI for Internet applications. The current revision of the URI specification, plus ongoing debate within the IETF and W3C communities on several proposed URI specifications, have delayed the processing of this Draft. DOI implementation does not depend on implementation of this specification.
URN implementation

The URN (Uniform Resource Name) specification is RFC 2141 URN Syntax. In practice, the URN specification defines (1) a formal registration process as a urn namespace, e.g., urn:doi:10.1000/1 and (2) accompanying specifications to implement a series of functional requirements for such namespaces.

- Namespace referencing. One may specify any existing identifier as a URN: e.g. urn:isbn:123456789, but this has no advantage over the simpler isbn:12345678. Such identifiers may be implemented using a specially written URN plug-in and resolved to URLs: functionally this gives nothing beyond the functionality achieved by coherent management of the corresponding URLs.

- URN implementation. In order to implement the functional requirements, the URN architecture assumes an additional network service: a DNS-based Resolution Discovery Service (RDS) to allow a client to deal with a previously unknown URN type by finding the specific service appropriate to the given URN scheme. URN resolutions are then delegated to that scheme-specific resolution service. However, no such deployed RDS schemes currently exist: browsers cannot action URN strings without some additional programming in the form of a “plug-in”. The lack of any widespread infrastructural support will require any URN implementation to develop their own resolution mechanisms, such as plug-ins or proxy servers. Resolution mechanisms which require functionality beyond 1 URN to 1 URL also require the creation of data models. Several such implementations have been developed for specific uses where deployment to a closed group of users may be achieved; these carry no guarantee of ready interoperability with other deployments, which may require a different plug-in for each implementation and may use conflicting data approaches.

- DOIs do not require a plug-in but offer this as an option. The Handle browser plug-in (native resolver, available in binary) delivers URN functional requirements in Windows-based browsers. URN plug-ins and Handle plug-ins share the problem of any new functionality of deploying the software to users. Handle has significant advantages: (1) the Handle system is a global supported resolution service; (2) the Handle browser plug-in is freely available, widely tested and proven across multiple applications; (3) unlike URN plug-ins, it is part of a suite of freely available and managed supporting software configured to provide coherent server-side support, including the Local Handle System Server and Handle System Client Library. These are available across platforms and with several added security features such as trusted resolution and distributed administration. DOI functionality can therefore be delivered through http, browser plug-in, or incorporation of handle software in a dedicated application.

- DOI is not registered as a formal URN, despite fulfilling all the functional requirements, since URN registration appears to offer no advantage to DOI. It requires an additional layer of administration for defining DOI as a URN namespace (the string urn:doi:10.1000/1 rather than the simpler doi:10.1000/1) and an additional step of unnecessary redirection to access the resolution service, already achieved through either http proxy or native resolution. If RDS mechanisms supporting URN specifications become widely available, DOI will be registered as a URN.
DOI functional requirements

The DOI system is designed to fulfil several additional functional requirements which we believe offer significant advantages in generic naming, notably:

- Neutral as to implementation. DOI allows but does not require http or other protocols. The design principle is that DOIs are not specific to the web or any other implementation (e.g. information may be delivered in non-web platforms such as PDAs). DOI is designed to be applicable in any environment on the Internet (the global information system linked by a globally unique address space based on the Internet Protocol (IP) using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite).

- Granularity of naming and administration at the object level. Allows but does not mandate coarser level granularity tools such as domain names. Specifically, DOI resolution in native resolver form does not require the use of the DNS (Domain Name System): the DNS administrative model argues against using it as a general-purpose name system and has well-recognised problems of security and updating.

- Neutral as to language/character set. Compatible with, but not restricted to, the ascii character set. DOIs can use the Unicode capability of the Handle system to develop DOIs in Japanese, Chinese, etc characters. The current DOI syntax restricts initial implementations to ascii simply for ease of adoption, but is intended to be widened (backward compatibly) to Unicode at the next revision.

This is one of a series of DOI factsheets. To see the latest version of this factsheet online, and to see the other factsheets:

- DOI and Handle
- DOI and Numbering Schemes
- DOI and Data Dictionaries
- DOI Applications

go to: http://www.doi.org/factsheets.html