the national archives

Digital Preservation Technical Paper:

2

The PRONOM PUID Scheme:

A scheme of persistent unique identifiers for representation information

Document Control

Author:	Adrian Brown, Head of Digital Preservation
Document Reference:	DPTP-02
Issue:	2
Issue Date:	27 July 2006

Contents

1 Introduction

This document is the second in a series of technical papers produced by the Digital Preservation Department of The National Archives (TNA), covering detailed technical issues related to the preservation and management of electronic records. This technical paper describes the scheme of persistent unique identifiers for representation information developed by TNA as part of its PRONOM system.

PRONOM is a technical registry service which describes the technical dependencies of digital objects in order to support long-term preservation. These technical dependencies, which correspond to the OAIS concept of representation information, include the formats in which objects are encoded, the software tools which may be required to perform actions on those objects (such as creation, rendering, and migration), and the operating system and hardware dependencies of those tools.

The PRONOM Persistent Unique Identifier (PUID) is an extensible scheme for providing persistent, unique and unambiguous identifiers for units of representation information recorded in the PRONOM registry. Such identifiers are fundamental to the exchange and management of digital objects, by allowing human or automated user agents to unambiguously identify, and share that identification of, the representation information required to support access to an object. This is a virtue both of the inherent uniqueness of the identifier, and of its binding to a definitive description of the representation information in a registry such as PRONOM.

At present, the PUID scheme has been confined to one particular class of representation information: the format in which a digital object is encoded. Formats are considered a particular priority for such a scheme. No existing, universally-applicable system provides for this. Unix ‘magic numbers’ and Macintosh data-forks do provide some of this functionality, but the same is not true within DOS or Microsoft Windows environments. The three-character file extension is neither standardised nor unique, and is interpreted differently by different environments. Equally, the IANA MIME-type scheme does not provide sufficient granularity or coverage to satisfy the requirements for unique identifiers. The PUID scheme has been developed for the single purpose of providing such identifiers.

The scheme is designed to be extensible, and will be expanded in future to include other classes of representation information in PRONOM, such as compression methods, character encoding schemes, and operating systems.

2 Scope

The PUID scheme has been developed in accordance with the following criteria:

- Uniqueness: Each PUID must be unique to a single unit of representation information, such as a specific version of a file format.

- Persistence: Once assigned, PUIDs must be persistent. As such, they must be immune from changes in technology or scheme administration. The scheme must be sufficiently flexible to be adaptable to future developments without any need for *post facto* changes to existing identifiers.

- Ubiquity: The PUID must be technology independent, and capable of describing any class or granularity of representation information.

- Focus: The PUID should not be used to convey any information beyond that required for identification.

- Brevity: The PUID should be as concise as possible, subject to the requirements of the other criteria.

The PUID scheme is designed to be applicable to any class of representation information capable of being described within the PRONOM registry. However, at present, its implementation has been limited to a single class: file formats. Within the context of this scheme, a file format is defined as a follows:

> *The internal structure and encoding of a digital object, which allows it to be processed, or to be rendered in human-accessible form. A digital object may be a file, or a bitstream embedded within a file.*

This structure and encoding will usually be formally expressed as a technical specification, although *de facto* standards also exist without formal specifications, such as Comma Separated Variable (CSV) format. File formats may be software-independent, or developed in tandem with specific software products. Format specifications are subject to regular revision, resulting in new format versions.

The granularity at which separate formats are identified is a crucial feature of the scheme. The PUID identifies formats at the most specific possible level of granularity. For example, the eXtensible Markup Language (XML) is a format which exists in a number of different versions (currently 1.0 and the forthcoming 1.1). Each version is regarded as a distinct format within the scheme. The Scaleable Vector Graphics (SVG) format is both a separate format in its own right, with three versions (1.0, 1.1 and the forthcoming 1.2), and an XML format. Each SVG version has its own specification, which makes reference to, but is distinct from, the XML specification. Thus, each SVG version is also distinguished by a separate PUID.

However, the granularity of PUIDs extends only to features which separate one format from another, and not to those which are inherent to a format. For example, the TIFF 6.0

image format supports a number of different image compression algorithms (RLE, CCITT Group 3 and 4, JPEG etc.), but these all relate to a single format.

In particular, the PUID does not distinguish on the basis of the following:

- Character encoding schemes: File formats may use a variety of different character encoding schemes, such as Unicode UTF-8 or US-ASCII. In most cases, the allowable encoding schemes are defined as part of the specification; as such, they are not elaborated within the format PUID. However, an additional class of PUIDs for identifying character encoding schemes will be implemented at a future date.

- Byte orders: Most formats use a specific byte order, either defined within the specification or as a consequence of the operating system within which they are created. Some formats, such as TIFF, support multiple byte orders. However, differences in byte order are not distinguished within the PUID.

- Encapsulated formats: Some formats support the encapsulation of other formats (e.g. TIFF images embedded within a PDF file). The PUID does not itself distinguish this. However, the PUIDs for both container and encapsulated components can be cited to support the modelling of such relationships within a metadata management system.

- Classifications: The PUID does not incorporate any form of classification system – such schemes are largely subjective and many formats do not lend themselves to simple categorisation.

- Relationships: The PUID does not seek to express any relationships between formats, such as sub-type or super-type.

It must be recognised that the function of the PUID is not to describe the features of a particular instance of an electronic object in a given format, nor to provide the information required to perform any particular action on that object; the PUID is simply required to provide a persistent and unambiguous binding to a definitive description of that format (provided, for example, by PRONOM or another technical registry).

3 PUID Structure

A PUID is composed of two elements, the PUID type, and the actual identifier. The PUID type identifies the class of representation information to which the identifier refers, where each identifier is unique within that class. Thus, both elements are required to construct a unique identifier.

The syntax may be expressed in formal BNF notation as follows:

```
<puid> ::= <puid_type> ‘/’ <identifier>

<puid_type> ::= <token> | ‘x-‘ <token>

<token> ::= <Any PUID type defined and approved by TNA>

<identifier> ::= <fragment> | <identifier> <fragment>

<fragment> ::= <digit> | <letter>

<letter> ::= ‘a’ – ‘z’

<digit> ::= ‘0’ - ‘9’
```

The PUID type is expressed as a lowercase alphanumeric string. The following PUID types are currently defined:

PUID type	Description
fmt	File format

Other types may be defined in the future. The identifier itself is expressed as a lowercase alphanumeric string of arbitrary length. The PUID type and identifier are separated by a forward slash.

An example PUID for a file format would therefore be expressed as follows:

```
fmt/42
```

PUID types prefixed by ‘x-‘ are used to provide temporary, private or experimental namespaces for that type. These may be used, for example, where a system requires a PUID identifier to be present which has not yet been formally assigned. Thus, format PUIDs of the type ‘x-fmt’ might be assigned for formats which have not yet been assigned an ‘fmt’ identifier. An ‘x-‘ PUID should not be considered persistent.

4 Expression of PUIDs as *info* URIs

In order to allow PUIDs to be expressed as Uniform Resource Identifiers (URIs), and make those identifiers available for public use in Web-based description technologies, the PUID scheme has been registered as a namespace under the *info* URI scheme. This scheme has been developed by the library and publishing communities to facilitate the referencing by URIs of information assets which have identifiers in public namespaces but have no representation within the URI allocation. It provides a simple URI registration mechanism to support the referencing of public information assets in advance of any possible subsequent URI scheme or URN namespace application.

The PUID scheme is registered with the *info* URI registry using the namespace ‘pronom’. A PUID identifier can be expressed as an *info* URI as follows:

```
info:pronom/<PUID>
```

The example PUID given in Section 3 would therefore be expressed as follows:

```
info:pronom/fmt/42
```

Full details of the “info:pronom” namespace are available from the *info* URI registry, at http://info-uri.info/.

5 Use of PUIDs

It is anticipated that PUIDs will primarily be used within managed electronic information environments, such as Electronic Records Management Systems (ERMS) and digital repositories. As such, PUIDs will normally be automatically assigned to individual electronic objects in one of two ways:

- At the point of creation, either by the authoring software or the ERMS.
- At any subsequent point in the object’s life cycle, either as part of the ingest process for an ERMS or digital repository, or through the use of separate file format identification tools.

An example of a file format identification tool which supports PUIDs is TNA’s DROID (Digital Record Object Identification) software. This freely-available tool performs automatic identification of the formats of digital objects, and will report the corresponding PUID (where assigned) for all identified objects. DROID provides a command line API to support integration with ERMS or repository software, and the identification results can be returned as an XML file, to allow simple transformation into any metadata scheme.

Neither the PUID scheme, nor its expression as an *info* URI, support any inherent dereferencing mechanism, i.e. a PUID does not resolve to a URL. However, TNA is planning to develop a range of services to expose PRONOM registry content, including a resolution service for PUIDs.

6 Scheme Administration

The PRONOM service and namespace, and the PUID scheme, are administered by the Digital Preservation Department of The National Archives. PUIDs are only authorised and assigned by TNA, and disseminated via the PRONOM registry (http://www.nationalarchives.gov.uk/pronom/).

The coverage of the PUID scheme is intended to be as extensive as possible, but with a current emphasis on file formats which may be used to create and store electronic records. New PUIDs are assigned on a regular basis, both to reflect new developments and to extend coverage of existing formats. In order to support the persistence and consistency of the scheme, new PUIDs are only assigned after thorough research and evaluation, to ensure that all variants of a given format are sufficiently understood to assign new PUIDs with confidence.

Requests for new PUIDs are particularly encouraged from those responsible for the management of ERMS, other digital preservation facilities, and from software and format specification developers. Submissions of supporting technical information on file formats and software tools for inclusion on PRONOM are also welcomed. TNA also welcomes external requests for the assignment of new PUIDs. Requests should be submitted by email to pronom@nationalarchives.gov.uk. Upon receipt, TNA will evaluate the request, undertake any necessary research, and assign a PUID as appropriate. The submitter will be notified of the result of the request by email, within 10 working days, although the time required to research and assign a new PUID, and to publish it via the registry, will be dependent upon the availability of the necessary technical information, and the complexity of the format.