About Data Formats

This page gives background information and describes the issues that
are addressed in the BitstreamFormat Renovation project page.
It is also a manifesto of sorts that presents some of the reasoning behind
the new format architecture.

About+Data+Formats

As we discuss it here, a "data format" is the description of how
intellectual content (i.e. the abstract meaning) is encoded in
a digital byte-stream to form what we call a *digital object*.
The format description is *technical metadata*, since it describes
how e.g. software would interpret the contents of a Bitstream to
deliver the intellectual content.

In Automatic Format Identification using PRONOM and DROID,
Adrian Brown defines a "data format" succinctly as:

The internal structure and encoding of a digital object,
which allows it to be processed, or to be rendered in human-accessible
form.

Why Data Formats Matter

The purpose of a digital archive or repository like DSpace is to find
and deliver digital objects for users. However, the real goal of the
users is to get at the intellectual content encoded in those digital
objects. Providing a dissemination format that can readily be
interpreted by common desktop applications and *accurately naming it*
is an integral part of the job. If you doubt that, try changing the
BitstreamFormats of an Item's contents to *Unknown* and see how useful it is.

Now that we have established that some knowledge of data formats and
technical metadata is necessary to fulfill DSpace's mission, we'll
break down the various uses of technical metadata, and show
how much there is to know about data formats.

Not all uses of formats are relevant to every DSpace installation. For
example, preservation of digital objects has specific demands of formats,
but not every DSpace administrator is concerned with preservation.
However, the architecture has to support preservation activities
for those users that need them (and others who may be surprised at how
relevant preservation is to them after all).

What Goes Into a Data Format

Typically, the description of a format may include:

- A descriptive name, e.g. `"OpenDocument Format"`
- Identifiers, including:
    - Globally unique and persistent identifiers from a format *registry*, e.g. the PRONOM PUID `fmt/135`
    - Application-oriented identifiers such as the MIME type, e.g. `text/html`
- Rigorous definition of the format's encoding, and documentation showing how to interpret it.
- References to other formats upon which it is built, *e.g.* XML depends on the ASCII (or UNICODE) character set and UTF-8 encoding.
- Relationships to other formats as subtypes, families, etc.
- References to application code that generates or interprets the format.
- Classification descriptive metadata about the nature of the intellectual content or encoding.

Not all of this metadata about formats is relevant to the mission of DSpace.
And, fortunately, there are public *format registries* such as
PRONOM which are
already collecting and maintaining it.

DSpace Use of Data Formats

What does DSpace do with formats, and what *should* it be doing?
Each kind of use has its own requirements of the format technical metadata:

Dissemination

When disseminating a Bitstream e.g. through a Web-based UI, DSpace needs a MIME type for format technical metadata. The only problem with this is that MIME types are poorly standardized, so there might be several valid identifiers for the same format. The recipient of the dissemination can only handle certain MIME types, so if DSpace gives it an unfamiliar one it will not render the content correctly.

To correct problems of patron's browsers not recognizing the MIME types that DSpace sends, it may be more practical to adjust DSpace than the browser. This is possible (and has been done) now, in DSpace 1.4.

Search

Does anyone search on actual Bitstream formats? The DC *type* element is similar to a format but not quite the same. Has anyone configured DSpace to put the types present in member Bitstreams into a search index for Items (or wanted to)?

The search criteria would probably be formats at a *very* coarse-grained level, e.g. "image", "audio", "text".

Locating Applications

DSpace uses format metadata to locate applications relevant to a Bitstream. For example, the `MediaFilter` mechanism processes a Bitstream by getting its format, and looking for filters that accept that format as their input.

This depends on the format of the Bitstream being accurately and completely identified. If a Bitstream's format was set to the wrong one or is completely unknown, it will not be processed.

There must also be a way to describe the formats each application accepts.
This requires describing ranges of acceptable formats in the DSpace configuration.
If we have thousands of formats available from a registry, it ought to also offer some simpler means of describing the range of formats accepted by an application, ideally without listing every individual format. For example, if formats are modeled in a hierarchy or by family groups, perhaps a range of fine-grained formats can be indicated by naming its superclass or parent.

Preservation

Data formats are also critical to all digital preservation operations:

Validation

To validate whether a Bitstream conforms to its identified format, it is first necessary to know its format in precise (fine-grained) detail. For example, if its format has distinct versions, the particular version (e.g. "PDF 1.2") must be identified, so the validation is meaningful.

Obsolete Format Detection

This also requires fine-grained format identification, because some versions of a format within a family of formats will usually go obsolete before the rest, so formats must be identified down to the version.

Migration

Migration is very similar to the existing `MediaFilter` application, since it is done by a collection of filters that translate obsolete or unpreservable formats into formats more suited to digital preservation.

A fine-grained identification of formats is necessary so we can choose only the
Bitstreams in immediate danger of obsolesence, but then there has to
be a way to match that specific format against a possibly-coarser-grained
declaration of the formats accepted by a migration tool.

Why the DSpace Format Model is Inadequate

The data model and implementation of `BitstreamFormat` was
originally intended as a placeholder to be supplanted by
an externally-developed format registry such as
the GDFR. Unfortunately,
progress has been slow in the field of format technical metadata,
so there hasn't been any obvious need to revisit the original design
decision because of the ultimate format registry coming on the scene.

There are *some* external format registries available now, however, and
a prototype implementation of the GDFR itself is under active development.
Since the FACADE project is dedicated to
improving DSpace's digital preservation capabilities,
resources are also available to upgrade DSpace's use of data formats.

Here are some particular issues to be addressed, explained
in greater detail below:

- `BitstreamFormat` short names are meaningless (as standard identifiers) outside of DSpace.
    - Difficult to use external preservation tools without common format names.
- No fine-grained format representation.
- Lack of technical metadata useful in preservation (format documentation, links to tools, etc.)
- Cannot leverage format work done for other format registries.

Granularity of Data Format Technical Metadata

The *granularity* of a data format definition refers to how broad or
narrow its concept of the format is. A *fine-grained* format description
is limited to a particular indivisible version or subset of a format,
e.g. "PDF 1.2". A *coarse-grained* format encompasses an entire family
of formats, *e.g.* "PDF" (all versions and subsets), or "PostScript".

DSpace 1.4 includes descriptions of fewer than 40 coarse-grained formats.
Also, its method of identifying formats is crude and extremely
vulnerable to error and failure, since it depends on recognizing filename
extensions of uploaded content files.

A serious preservation effort demands fine-grained indentification of
file formats, and more reliable format identification.

We acknowledge that not every DSpace installation is required to
implement preservation on its contents, so it will still be possible
to configure simpler and coarser format identification tools.

Note that coarse-grained format descriptions can be useful as well,
so long as there is a way to discover the fine-grained formats that conform to each coarse format.
It is easier to cite a coarse-grained format when specifying the
formats acceptable to an application, or when searching for Items
with components of a given format.

Data Format Identifiers

Identifiers are names for data formats. There is obvious value in
having a set of identifiers that:

- Are globally recognized, can be exchanged meaninfully between Dspace, other archives, and applications.
- Each uniquely identify a data format, i.e. correspond 1:1 with formats.
- Indexes an entry in a publically-available format registry, which contains metadata about the formats.

What is Wrong With MIME Types

Many of today's most popular multimedia applications, such as Web browsers
and email user agents, recognize
MIME types
as data format identifiers. I believe they employ MIME types because
they are already widely implemented as part of the mail system, not because
they are a good solution.
In practice, MIME types often fail to get
content rendered correctly, requiring adjustments and human intervention.

Problems with MIME Types:

1. There is no authoritative standard that actually gets followed, resulting in multiple "valid" identifiers for the same format (e.g. `model/iges` and `application/iges` have both been officially recommended at various times; *unofficial* usage is all over the map.)
    - IANA attempts to provide a central registry.
    - Applications receiving MIME types are wont to be configured with the types they commonly receive, regardless of the "standard".
2. The same format can be legally described at different levels of granularity, e.g. these are *all* valid descriptions of RDF/XML:
    a. `Content-type: application/rdf+xml` *(specifies RDF encoded as XML)*
    b. `Content-type: text/xml` *(only acknowledges the XML level of encoding)*
    c. `Content-type: application/xml`
3. In practice, it is never used for fine-grained format identification. Each format has its own rules (if any) to distinguish versions within a MIME type.

It's worth noting that Apple Computer felt it necessary to
invent their own Uniform Type Identifiers
to take over the role of MIME types on MacOS X. The UTIs are
hiearchical, and cleanly represent families of formats and various granularities.
They give each vendor (or organization) a private namespace to control,
for unambiguous extensibility. It is *not* a file format
registry, however, since there is no metadata, just names.

DSpace Abuse of "License" Format Name

The current (1.5) DSpace data model applies the special
BitstreamFormat named "License" to all Bitstreams in the "LICENSE" bundle.
This is a *convention* in the data model that identifies the Bistreams
as, effectively, *rights metadata*. It has two serious flaws:

1. Because it overloads the purpose of the BitstreamFormat metadata, there is no way to describe the actual data format of the License Bitstream.
2. It is unnecessary and redundant, since all License Bitstreams are stored in the "LICENSE" bundle anyway.

Here is a case study that clearly illustrates the hazard of stealing a
Bitstream's technical metadata:
In the
course of moving from a Windows to Unix platform,
this DSpace administrator has discovered
a digital preservation problem.
Many existing License Bitstreams need to be either converted in place or
labelled correctly on dissemination so they are understandable. If they
had honest BitstreamFormat metadata accurately identifying their format,
it would be straightforward to pick out the problem Bitstreams. As it is,
there may be License bitstreams representing XML and URIs for CreativeCommons,
so any mechanism to detect or fix them has to identify the file format
from the contents of the Bitstream. Also, given the existing format model,
the only way to repair this problem is to rewrite the License Bitstream in
the new preferred character set – which is *also* undocumented!

```
From: Christian Voelker &lt;C.Voelker@gmx.net>
To: "dspace-tech@lists.sourceforge.net Tech" &lt;dspace-tech@lists.sourceforge.net>
Date: Fri, 1 Feb 2008 23:03:07 +0100
Subject: [Dspace-tech] How to change license.txt bitstream for all items

Hello,

we have moved from one server to another and on the new
machine, apache is configured to deliver text files with
the proper utf-8 encoding. On the former machine, this
was obviously not the case. The former admin decided to
save the license.txt in windows encoding which made it
display properly in the users browser then. But now, it
looks really nasty. Example:

http://www.stadtteilgeschichten.net/retrieve/940/license.txt

Now I found, that all these licenses get copied to the
assetstore at the time they are granted with a timestamp
and username in the first line making each of them unique
and an associated database entry containing the precise
file size.

Now, I could certainly put together some script to find
all license files and change the special characters.
This would probably reduce the file size as well, which
would introduce errors to the db implicitly. Ok, I could
produce another script to change the database entries
accordingly. All this looks like quite some work for
a more or less cosmetic problem. I see that the design
makes sense because content should not be changed in an
archive after the date of ingestion, but I think that
this change is not dishonest.

I could also reconfigure apache to deliver text files
the old way but this means to perpetuate the situation
for all items added from now on.

Any advice appreciated.

Bye, Christian
```

Value of External Format Registries

There are many advantages to using a data format registry outside
of DSpace as the main source of format technical metadata, so long
as it is stable and well-maintained.

- Formats have globally-unique, persistent identifiers
    - Some other application talking to DSpace can refer precisely to a format with this external identifier, so it is mutually understandable.
- It provides technical metadata (and possibly tools) describing how to identify formats by internal and external characteristics.
- Includes references to standards documents and other descriptions.
- Registries typically include descriptive metadata showing relationships between formats, allowing fine-grained definitions to be grouped under coarse-grained formats.
- A central registry gets support and attention from other users, which we can leverage.
    - Adding documentation of new formats
    - Catching and correcting mistakes
    - Monitoring formats for obsolescence