Internet-Draft
<draft-carrasco-xdossier-04.txt>
Expires 14 December 2004

M.T. Carrasco Benitez
Dragoman
15 June 2004

Xdossier

Status of this memo

Abstract

This is an informational memo for Xdossier. A Xdossier is a data object designed for browsing with web browsers and mappable to XML. It is based on a directory structure containing files in several formats.

Table of Contents

1. Introduction

It is recommended to play with a Xdossier example, as this memo should be easier to understand. For examples look in http://dragoman.org/xdossier.

This recommendation is about organising files. They are organised into a data object called Xdossier.

Informally, a Xdossier is a directory structure with files in several formats created for web browsing; direct browsing ("file:") or served browsing ("http:").

Classifying files within directories is easy and very instinctive. A few HTML files with some descriptions and links can greatly help the browsing and give a feel of "oneness". One can easily start organising using the directory structure point of view. By following a few rules,

one can end up with a data object easy to browse and with a significant structure.

A directory structure is a tree similar to an XML document. There is a strong parallelism. With a formal mapping to XML, the directory structure could be transformed into an XML document.

One could start with the structure of the directories and files (the "Backbone") and progress with the structuring towards the content of the individual files (the "Leaves"): a few files could be XML files, eventually the whole Xdossier should be transformable into a XML document.

This approach is particularly useful to organise large amount of legacy data in several formats for which there is no clear formal definition.

2. Rationale

- Usable with web browsers. At most, only unpacking (e.g., unzipping) should be necessary.

- Easy to "produce" and easy to "consume".

- Usable "as is" and adapted to further processing. For example, a CD-ROM must be usable directly ("raw" consumption) and programs should be capable of mechanical processing to load into a DBMS, web server, etc.

- Easy to prepare with resources (computer equipment, programs, staff, etc) in most firms or acquirable at low cost. In particular, it should be easy to prepare by hand without the need of special programs.

- Mappable to XML [XML].

- Vendor independent.

- Usable as an interface to exchange data.

3. Terminology

The specific terms to this memo have usually the first character of each token in capital. Many term and concepts are the same or parallel to SGML/XML and file systems.

- Index: Abbreviation of "Node Index".

- Instance: Abbreviation of "Xdossier Instance".

- Minimal Root: Abbreviation of "Minimal Root Xdossier".

- Minimal Root Xdossier: Xdossier with a minimal number of elements in the Root Node.

- Node: Abbreviation of "Xdossier Node".

- Node Index: File, usually named "index.html" that contains links to and information on files in a particular Node.

- Node Store: An optional directory named "xdossier" that could be present in each Node.

- Root Index: The Index in the Root Node.

- Table of Contents: Abbreviation of "Xdossier Table of Contents".

- Templated: Abbreviation of "Templated Xdossier".

- Templated Xdossier: A Xdossier constructed following the indications of a Xdossier Template.

- Xdossier: (1) The concept as described in this memo. (2) Abbreviation of "Xdossier Instance".

- Xdossier Instance: Parallel meaning with XML document instance.

- Xdossier Node: A directory and his components.

- Xdossier Table of Contents: The Root Index. The Xdossier Table of Contents must allow the navigation of the whole Xdossier. Typically, there would be links to other Directory Indexes.

- Xdossier Template: A Xdossier that indicates how to construct similar Templated Xdossiers. I can be viewed as a "light" DTD.

4. Xdossier

A Xdossier is a data object composed of a directories/files structure that follows this specification. In particular, Xdossiers must follow the rules regarding names, representation, file extension, file format, character encoding and web format. The start is the Index in root directory.

4.1. Mapping between Xdossier and XML

The mapping between Xdossier and XML concepts is as follow:

Xdossier		XML
directory	<->	element
directory name	<->	element name
root directory	<->	document element
Index	<->	attributes (for his directory and files)
file	<->	entity
file name	<->	entity name
file content	<->	entity reference
XML file	<->	Parsed entity
Non-XML file	<->	Unparsed entity

This could be used for transforming between Xdossier and XML. Xdossier should be transformable into XML.

5. Xdossier types

There are three types of increasingly strict Xdossiers:

- Well-formed: All Xdossier must be well-formed as defined in this

specification.

- Templated: A Xdossier that in addition follows the indications of a Xdossier Template.

- Valid: A Xdossier that in addition follows the rules of a precise syntax; e.g., DTD, schema [SCHEMA], value-pair, etc.

The concepts of "well-formed" and "valid" are parallel to XML. "Templated" does not exist in XML.

5.1. Well-formed Xdossier

A well-formed Xdossier must follow the rules of construction in this specification. All Xdossiers must be well-formed. This is the minimum requirement to be a Xdossier. A well-formed Xdossier does not have to follow the additional indications/rules of a Template/syntax.

Rules of construction refer to the part of this specification covering aspects not related to Templated and valid Xdossier.

5.2. Templated Xdossier

A Templated Xdossier follows the indications of a Xdossier Template, abbreviated to Template. A Template is a Xdossier declared as a Template. Usually, Xdossier Templates would be purpose built Xdossiers to fulfil the role of Template.

The presence of directories/files in a Template would mean that they must be present in Xdossiers; usually with the same name and format. There can be additional indications, in particular the Indexes; e.g. "such a file is optional". Probably, some aspects could be fuzzy.

People with limited knowledge in computers could create Templates as it is instinctive. Probably, the path would be to create a well-formed Xdossier and then to proceed with the creation of a Template.

As the approach does not have a fixed syntax, it is not intended for full mechanical validation by computer. Some parts would have to be validated by humans, though parts that follow a syntax could be

validated mechanically. For example, the content model of the files/directories could be defined as:

- DTD: an XML DTD.

- Pair of values: for example a list of pair of values like "/food/choco/index.html=Documents about chocolate"

5.3. Valid Xdossier

A valid Xdossier follows the rules of a syntactic system such as DTD or schema [SCHEMA]. This is needed to implement computer programs that could do full mechanical validation of Xdossiers.

Another memo should address the syntax. Schema and XML DTD will be considered.

6. Web Formats

These are file formats well adapted to the web and supported by widely available browsers. A very good format for the web, but not supported by widely available browsers is not a Web Format.

Web Format is a fuzzy moving definition. It is also "community dependent"; e.g., a community could consider XML a Web Format and another community could consider that it is not a Web Format.

By default, the only Web Format is XHTML.

[Relation to Templated and valid Xdossier]: It could redefine the list of Web Formats.

7. Xdossier Node

A Xdossier Node, abbreviated to Node, is a directory and the following:

- Xdossier Node Name, abbreviated to Node Name; the name of the directory.

- Node Index.

- Node Store.

- File(s) in this particular directory.

- Name(s) (not the content) of the directory/ies in this particular directory.

8. Node Index

Node Index, abbreviated to Index, is a document in Web Format included in each Node. Indexes should/could have two functions:
- Browsing (informal view).
- Metadata (formal view).

The browsing and metadata are functions. Syntactically, they could be interwoven.

Syntactically, there are two types of Indexes:
- Informal Index: it does not follow any particular syntax.
- Formal Index: It follows a syntax.

If Index is not present, the filenames in the directory should be meaningful.

The default Document Name for Index is "index" and the default format is the default Web Format. Hence, at present the default File Name for Index is "index.html".

[Relation to Templated and valid Xdossier]: It could redefine the default Index name.

8.1. Browsing function

Indexes should have a human readable description of his Node and meaningful labels with links mostly to:
- His file(s).
- Indexes in child directories.
- Navigational aids (e.g., a link to the Root Index).

One should be able to view all the directories/files in the Xdossier starting from the Root Index and following links, except if the intention is to hide them. Hence, every directory/file should have a link pointing to it. Usually from his Index, but it could also be from other Indexes or files.

Nodes should be as self-contained as possible. Hence, it is recommended for Indexes to have links only to his files and child Indexes; i.e., the Indexes of his directories. Though, Indexes could also contain links to other files/resources.

Links to files within a Xdossier must be relative.

8.2. Metadata function

Indexes could contain the metadata of his Node. The metadata should be machine processable. The metadata could also be in the Node Store.

Another memo should address the metadata. Resource Description Framework [RDF] will be considered.

9. Node Store

Node Store, abbreviated to Store, is an optional directory that could be present in each Node. If it is present, it must be named "xdossier"; this name is reserved for this purpose.

The Store could contain additional data related to the Node where it is situated. For example, metadata for his directory/file(s), previous versions of the directory files, etc.

The Node Store in the root directory is called the Root Node. The Root Node could contain the information to make the Xdossier Templated or valid. This is similar to a DOCTYPE in an XML document and the DTD.

Another memo should address specifications of Store.

10. Root directory

The Node Index in the root directory is called the Root Index. The root directory must contain only one file, the Root Index; and zero or more directories. Corollary: The trivial Xdossier is composed only of the Root Index.

The intention for allowing only one file (the other elements must be directories) in the root directory is to make it obvious that the file present (the Root Index) is the Table of Contents.

It is recommended to minimise the number of elements in the root directory, or at least to keep it to a reasonable number.

Minimal Root Xdossier, abbreviated to Minimal Root, is when the Root Node contains only the Index, one directory and optionally the Store. The intention is to make it even more obvious for the user. Minimal Root is appropriate for Xdossiers not intended for loading into web servers, as the URLs are longer.

11. Self-containness

There are three levels:

- Absolute Xdossier: When all the resources are in the Xdossier.

- Self-Contained Xdossier: When all "Essential Resources" are in the Xdossier. For example, the CSS is in the Xdossier, though there could be secondary references to other resources such as a reference to the W3C site at http://w3.org. At least this level should be attained.

- Fragment Xdossier: When at least one "Essential Resource" is not in the Xdossier. For example, the CSS is not in the Xdossier and it relies in an external CSS such as the one in the W3C site at http://www.w3.org/StyleSheets/Core/. It is only recommended as a directory of Xdossier. Otherwise, there should be an agreement between producers and consumers of the Xdossier.

Essential Resources are the ones needed for navigation and display.

[Relation to Templated and valid Xdossier]: It could include the minimal level of Self-containness requested and a re-definition of the Essential Resources.

12. Compound Xdossier

It is a Xdossier where all the directories in the root directories are Xdossier themselves. These directories could also be Compound Xdossiers and so on.

[Relation to Templated and valid Xdossier]: It could include required Compound Xdossier.

13. Backbone

The Backbone is the directories and files names; i.e., the main branches of the tree. The Backbone is not concerned with the structure of the files.

There are two types of Backbone Formats:
- File System Backbone Format: a directories/files structure.
- Pack Backbone Format: a packed File System; e.g., zip.

File System Backbone Format is abbreviated to File System Format or simply File System. Pack Backbone Format is abbreviated to Pack Format or simply Pack.

The main difference is that today Pack must be unpacked before viewing with browsers. This could change if browsers could support Packs such as zip. For example, one could have:

```
file:///mydirectory/myfile.zip/index.html
```

or

```
zip:///mydirectory/myfile.zip/index.html
```

This should extract the file "index.html" from the zip file "myfile.zip" and display the content of "index.html" as if it is reading from a file system. Pressing the links pointing to other files in "myfile.zip" should behave in a similar fashion.

13.1. File System

These are directories and files as in a file system; e.g. Windows or Unix. Xdossier uses mainly the tree properties of file systems. Xdossier does not consider other properties of file systems such as access control list (i.e., the bits protection, ownership, etc) or links within the file systems itself (e.g., symbolic links). It is up to the user to set the correct access control list; e.g., to reset the executable bit in the appropriate files. Future versions of this memo should address this issue.

File System is more adapted to media such as CD-ROMs where one wants the Xdossier ready for use without any intermediary processing.

The File Systems in order of preference are: Joliet, others.

[Relation to Templated and valid Xdossier]: It could redefine the File Systems.

13.2. Pack

A directory structure could be packed into one or several file(s); e.g., zip. Packing must respect the directories/files structure. If a packing technique compresses, it is just considered a bonus.

Packed is better adapted for:

- Attaching Xdossier(s) to emails.

- In file systems that do not support the naming of the directories/files (it could easily happen with DOS).

- With large collections of Xdossiers that could cause problems in the

files system.

Care must be taken to unpack in a computer system that supports the naming in the directory structure. For example, name lengths of the directories/files and file extensions.

Another approach would be not to unpack the directory structure and view it with browsers that directly support the unpacking technique, as described above.

In the future, others aspects would be addressed: Xdossier that expands several Packs (e.g., pack1.zip, pack2.zip); mixed Pack Xdossiers (e.g., pack.zip and pack.tar); mixed File System and Pack (e.g., the Root Node as File System and the rest as Packs).

The Packs in order of preference are: zip, tar, cpio and others.

[Relation to Templated and valid Xdossier]: It could include a list of accepted packing techniques in order of preference.

14. File formats

Priority should be given to file formats with a good chance of being readable "forever"; e.g., in 50 years. This points to "neutral" formats: formal standard, industrial standard, vendor independent, "text-like", etc.

One should not disregard proprietary formats, as they could be the "source" format; i.e., the format in which the data was originally produced. Often, information is lost in format transformation. The recommendation is to include:

- A file in the source format.

- A file in at least one neutral format.

- Indicate the method used in the format transformations; e.g. source format saved HTML using the "Save as" facility in such application.

The file formats in order of preference are:

- Text: XML*, XHTML, HTML, XML, text, RTF, PDF and others.
- Graphic: SVG*, PNG or JPEG, GIF, TIFF and others.

*Future directions: XML will be the preferred format (text and graphic) when it is well supported by widely available browsers. At present, it is recommended to use as much as it is reasonably possible. It is recommended to use the appropriate XML applications such as Chemical Markup Language [CML].

The choice of formats is also dependent on the intention of the user; e.g., when giving preference to PNG or JPEG.

When other formats are used, they should be widely used formats; e.g. Word. Some could be widely used in a specialised field; e.g., SAS. In addition to the proprietary formats, it is recommended to include transformations to text-like formats with as much information as possible. For example, word-processing documents could be transformed into RTF and database tables into "comma separated" files.

[Relation to Templated and valid Xdossier]: It could include a list of accepted formats in order of preference and different mapping between the Internet Media Type and file extensions.

15. Representation

The same information could be represented in different fashions. The dimensions considered are:

- Language; e.g., English, Spanish.
- Media type; e.g., HTML, PDF.
- Encoding; e.g., zip, gzip, compress.

16. File extension

File extensions are used to indicate representations. For example:

hello	no extension
hello.html	format HTML
hello.en	language English
hello.gz	compressed using "gzip"
hello.en.html	English in HTML

```
hello.html.gz        HTML, gziped
hello.en.gz          English, gziped
hello.en.html.gz     English, HTML, gziped
```

File extensions, particularly the last one, are operating systems dependants:
- Syntax: e.g., DOS allows up to three characters file extensions.
- Association: which program is associated with the extension.

The extension should correspond to widely used mapping between Internet Media Types [IMT] and file extensions. The examples above work for Transparent Content Negotiation [TCN] in Apache [APACHE].

Note the difference between "file" and "document". File refers to physical storage; e.g., "mydoc.txt" is a file. Document refers to content; e.g., "mydoc" is a document represented in the files "mydoc.txt" and "mydoc.html", they contain the same document in different formats.

Another approach would be to use variants files (see TCN). Another memo should address the syntax for file extensions.

17. Character encoding

The character encoding ("charset") in order of preference are:

- Unicode UTF-8, Unicode 16 bits [ISO10646].

- ISO-8859-1 (Latin-1) or appropriate ISO-8859-x; e.g., ISO-8859-7 for Greek.

- Other character encodings. They should be appropriate to the language and widely available.

[Relation to Templated and valid Xdossier]: It could include a list of accepted character encoding in order of preference.

18. XHTML for Index

The XHTML used in the indexes should follow the rationale of XHTML Basic [XHTML-B]. Some indications:

- Simple mainstream XHTML; i.e., facilities easy to write and that work in most browsers.

- A link to the Root Index. One could also use the "Start" mechanism; e.g., "<link rel="Start" href="../index.html" />"

- It is recommended to use one CSS for all the Indexes.

- A reasonable presentation with the most popular browsers (e.g. Internet Explorer, Navigator, etc) and text only browsers (e.g. Lynx).

- Links that work when read directly (e.g., a CD-ROM inserted into a PC) or served by an HTTP server; i.e., "file:" or "http:".

- Links that point directly to files, except when the intention is to show the content of the directory. One should not assume that Xdossier would be served by server; i.e., it should work directly ("file:") or served ("http:").

- No frames, or at least a no frame option.

- No scripts (e.g. JavaScript) and Java Applets.

- Images (IMG) with alternative texts.

- Relative links within the Xdossier; e.g. href="../doc.html".

- Use language attributes (lang, xml:lang, etc), to indicate the language of the text.

[Relation to Templated and valid Xdossier]: It could change the XHTML indications.

19. Name

Xdossier must conform to the XML naming and respect that the name "xdossier", in all possible combinations of upper or lower case, is reserved; e.g., xdossier, XDOSSIER, Xdossier, xDossier, etc.

Xdossier are "Strict Name Conformance" when they also conform to section "19.1 Strict Name Conformance".

19.1 Strict Name Conformance

"Name" is a token composed of the following characters:
- Letters "a" to "z"; i.e., lower case only; [U+0061 to U+007A].
- Digits "0" to "9" [U+0030 to U+0039].
- "-" [HYPHEN-MINUS, U+002D].
- "_" [LOW LINE, U+005F].

The notation "U+" refers to the Unicode [UNICODE] notation.

```
Correct Names
  part_a
  part-b
  myfile
  hello
  xdossier-hello


Incorrect Names
  part a       (' ' ; SPACE is not allowed)
  Myfile       (capitals are not allowed)
  myfile.xml   ('.' ; FULL STOP is not allowed)
  hello:html   (':' ; COLON is not allowed)
  xdossieR     ('xdossieR' ; reserved)
```

"Directory Name" is a Name.

"File Name" is one Name followed by one or more Name(s) separated by a '.' (FULL STOP, U+002E).

```
Correct File Names
  a_part
  myfile.html
  hello.en.xml
  hello.en.xml.gz
```

```
Incorrect Names
  a part       (' ' ; SPACE is not allowed)
  Myfile.html  (capitals are not allowed)
  hello:xml    (':' ; COLON is not allowed)
```

"Document Name" is the first Name in the File Name. Example, "docname" in the File Name "docname.ext"

"File extension(s)" is/are the second and following Name(s). For example, "ext1", "ext2" and "ext3" in the File Name "docname.ext1.ext2.ext3"

20. References

[ALLEN] Package or Perish. Terry Allen
Pages 385-390 in SGML/XML '97 Conference Proceedings. SGML/XML '97.

[APACHE] The Apache Foundation
http://apache.org

[CML] Chemical Markup Language
http://xml-cml.org

[CSS2] Cascading Style Sheets, level 2
http://www.w3.org/TR/REC-CSS2

[DC] Dublin Core
http://purl.org/dc

[ESUB] Electronic Submission
http://esubmission.eudra.org

[ISO10646] Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane, ISO/IEC 10646-1:1993

[HTML] HTML 4.01 Specification
http://www.w3.org/TR/html4

[IMT] Internet Media Types
http://www.isi.edu/in-notes/iana/assignments/media-types/media-types

[MHTML] The MIME Multipart/Related Content-type. E. Levinson
ftp://ftp.ietf.org/rfc/rfc2387.txt

[RDF] Resource Description Framework Model and Syntax Specification
http://www.w3.org/TR/REC-rdf-syntax

[SCHEMA1] XML Schema Part 1: Structures ("work in progress")
http://www.w3.org/TR/xmlschema-1/

[SVG] Scalable Vector Graphics (SVG) 1.0 Specification (work in progress)
http://www.w3.org/TR/1999/WD-SVG-19991203

[TCN] Transparent Content Negotiation in HTTP
http://ietf.org/rfc/rfc2295.txt

[Unicode] Unicode Consortium
http://www.unicode.org

[XHTML] XHTML 1.0: The Extensible HyperText Markup Language
http://www.w3.org/TR/WD-html-in-xml

[XHTML-B] XHTML Basic ("work in progress")
http://www.w3.org/TR/xhtml-basic/

[XML] Extensible Markup Language (XML) 1.0
http://www.w3.org/TR/rec-xml

[XSL] Extensible Stylesheet Language Specification ("work in progress")
http://www.w3.org/TR/WD-xsl

21. Acknowledgement

The comments of Martin Bryan to an early draft were very useful. Also, he suggested the Template. As usual, the author is the sole responsible for the document.

22. Author

Manuel Tomas CARRASCO BENITEZ
Dragoman
Luxembourg

Telephone +352 26 200 747

xdossier@dragoman.org
http://dragoman.org/carrasco

23. Disclaimer

This document represents the view of the author. It does not necessarily represent the views of any other parties.