Australian Government
National Archives of Australia

# How Xena performs file format identification

**Version 1.0**

**RKS: 2009/4024**

the archives
naa.gov.au

# Document Change Record

| Version | Changed By | Description of Changes | Change Date |
|---|---|---|---|
| 0.1 | Allan Cunliffe | Created | March 2011 |
| 0.2 | Allan Cunliffe | Post internal review | March 2011 |
| 0.3 | Allan Cunliffe | Second review | April 2011 |
| 1.0 | Allan Cunliffe | Released | May 2011 |

## Related Documentation

| Title | Author | Date | URL |
|---|---|---|---|
| Dissecting the Digital Preservation Software Platform | Allan Cunliffe | February 2011 | http://www.naa.gov.au/Images/Digital-Preservation-Software-Platform-v1_tcm2-34756.pdf |
| An Approach to the Preservation of Digital Records | Helen Heslop<br>Simon Davis<br>Andrew Wilson | December 2002 | http://www.naa.gov.au/images/an-approach-green-paper_tcm2-888.pdf |
| The Benefit of Experience: the first four years of digital archiving at the National Archives of Australia | Michael Carden | August 2010 | http://michaelcarden.net/blog/wp-content/uploads/2010/08/michael-carden-conference-paper-final.pdf |
| Xena Help | Allan Cunliffe | August 2010 | http://xena.sourceforge.net/documentation.php |

# Table of Contents

# 1 Introduction

Xena[1] digital preservation software has been developed by the National Archives of Australia to aid in the long term preservation of digital records.

The main function of Xena is to determine the file format of digital records and convert them to an appropriate preservation file format based on open standards. This paper describes how Xena performs file format identification.

## *1.1 Audience*

This document is intended for anyone interested in the functionality of Xena and how it detects input file formats.

While no technical knowledge is assumed, the document deals with terms that may be unfamiliar. For more information about the technical terms used in this document, please refer to **Appendix B – Glossary**.

[1] http://xena.sourceforge.net/

# 2 Xena architecture

The main components of the Xena architecture are:

- Xena graphical user interface (GUI)
- Xena command line interface
- Xena object. The Xena object allows the basic functionality of the Xena system to be accessed, including:
  - loading plugins
  - guessing file types
  - normalising files
  - exporting normalised files
- Plugin Manager. The Plugin Manager loads the plugins and distributes the input source file to each of the plugins to determine the file format
- Plugins. Xena plugins consist of one or more components, each having a specific role in the conversion process (such as file format detection, file conversion and creation of the Xena XML file).

[2]

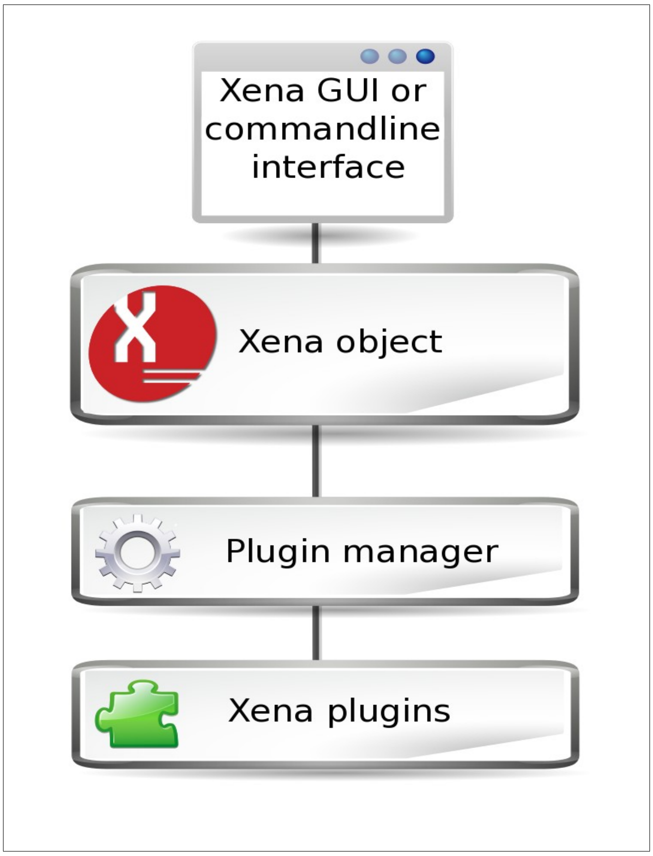

*Diagram 1: Xena architecture*

[2] Icon source: http://oxy-gnome.org/

# 3 Plugins

Xena uses different plugins to deal with various file types. The Xena plugins are loaded by the Plugin Manager when Xena is run.

## *3.1 Plugin structure*

Each plugin for Xena consists of a number of components. Each component performs a specific role.

Typical plugin components are:

- Xena Type – corresponds to a supported file format
- Guesser – determines the type of a given Xena input file
- Normaliser – takes the Xena input file and transforms it into an XML file
- De-normaliser – takes a normalised file, and transforms it back into a normal file[3]
- File Namer – names normalised and denormalised files according to a specific naming scheme
- View – displays Xena files.

For the purpose of file format identification, the most important components of the plugin are the Xena Type, Guesser and Normaliser. The main points to note are:

- There is a plugin for each logical grouping of supported file formats (such as email, image, audio, office[4]).
- Plugins contain one or more guessers and one or more normalisers.
- Plugins have a guesser for each file type they support.
- A single normaliser may normalise multiple file types.
- Each file type is normalised by a specific normaliser.

[3] For each de-normaliser, there is generally a single output target file format.

[4] See **Appendix A** for a list of the plugins.

The relationship between the plugin, normaliser and guesser are shown below, using the Office Plugin and office documents as an example:

[5]

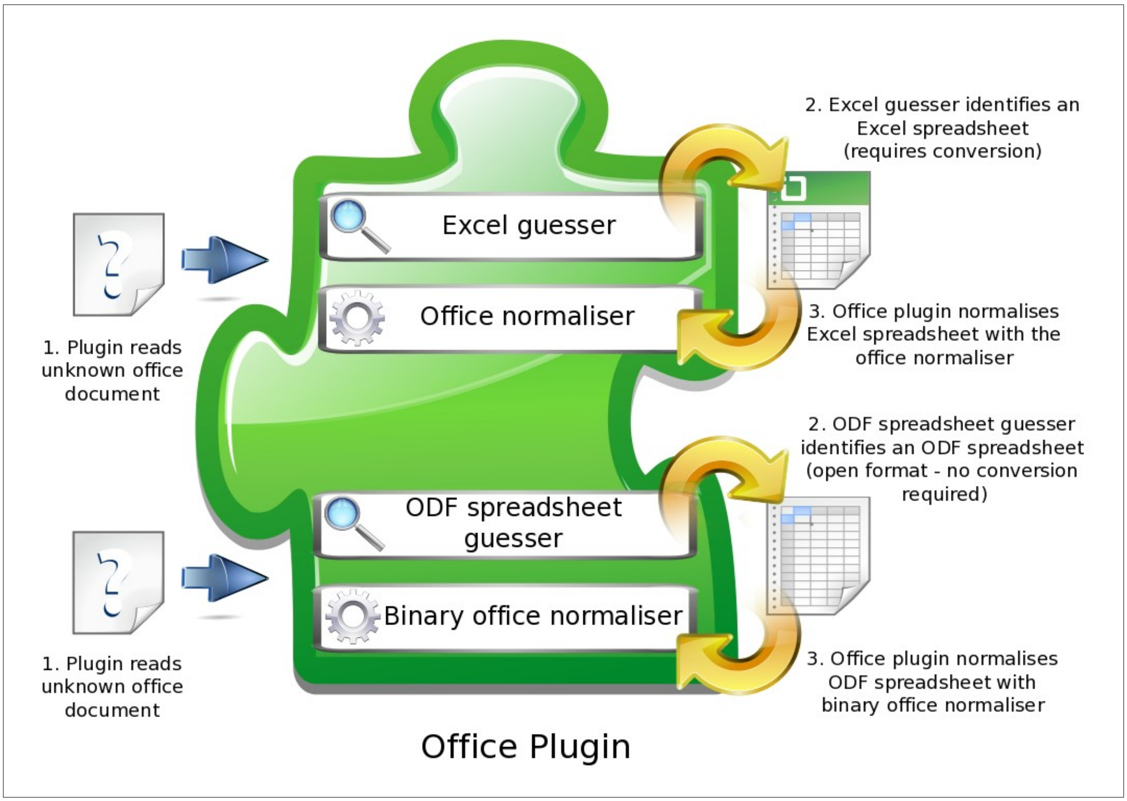
2. Excel guesser identifies an Excel spreadsheet (requires conversion)
Excel guesser
Office normaliser
1. Plugin reads unknown office document
3. Office plugin normalises Excel spreadsheet with the office normaliser
2. ODF spreadsheet guesser identifies an ODF spreadsheet (open format - no conversion required)
ODF spreadsheet guesser
Binary office normaliser
1. Plugin reads unknown office document
3. Office plugin normalises ODF spreadsheet with binary office normaliser
Office Plugin

*Diagram 2: An example of plugin functionality*

[5] Icon source: http://oxy-gnome.org/

# 4 Guesser

The guesser component of a Xena plugin is responsible for determining the file type of a Xena input file. Each guesser compares certain input file attributes with those stored in the guesser and the corresponding Xena Type.

Typical attributes used for comparison and their weightings are given in the following table.

| Attribute | Example | Weighting[6] |
|---|---|---|
| MIME type/header | MIME type of html files is *text/html* | • MIME_MATCH_FALSE = -10000<br>• MIME_MATCH_UNKNOWN = 0<br>• MIME_MATCH_TRUE = 60 |
| Magic number | JPEG image files begin with *FF D8* and end with *FF D9* | • MAGIC_NUMBER_FALSE = -10000<br>• MAGIC_NUMBER_UNKNOWN = 0<br>• MAGIC_NUMBER_TRUE = 50 |
| File extension | PDF documents commonly have the file extension *.pdf* | • EXTENSION_MATCH_FALSE = 0<br>• EXTENSION_MATCH_UNKNOWN = 0<br>• EXTENSION_MATCH_TRUE = 40 |
| Data in body of the file | Characters in the file are encoded according to the *UTF-8* character set | • DATA_LIKELY_FALSE = -30<br>• DATA_LIKELY_UNKNOWN = 0<br>• DATA_LIKELY_TRUE = 30 |

*Typical file attributes used to determine file formats*

To generate a score for each guesser, the scores for each attribute are added together. A high, positive score is more indicative of a particular file format than a lower score or a negative score.

For example, in the case of the JPEG guesser:

- if the input file has a JPEG magic number but no file extension, the guesser would set the file extension to 'unknown' and the magic number to ‘true’. Based on the table above, the guesser would give the file a score of 50.
- if the input file has a JPEG file extension but no JPEG magic number, the guesser would set the magic number to 'unknown' and the file extension to ‘true’. Based on the table above, the guesser would give the file a score of 40.

The difference between the two results above shows that a file extension match is not as important as a magic number match for file identification (as file extensions can be more easily altered than magic numbers).

Further examples of guesser functionality are given in **Appendix C**.

[6] Weightings are indicative only - depending on the guesser, other weightings may be applied.

# 5 File identification process

There is a guesser for every file format supported by Xena. Xena identifies the file type of an input file by checking it for certain file attributes:

- Each guesser compares the input file attributes with those stored in the guesser and the corresponding Xena Type.
- The result of the comparison is measured to give a likelihood of the input file being a certain file type.
- The file type profile that the input file most closely matches is the most probable file type.

The default file identifier is the binary guesser. If matches to other file types can not be established, the file is treated as a binary file type and normalised with the binary normaliser.

The following sections describe Xena's two-step approach to file format identification.

## *5.1 Initial identification attempt*

Passing input files to every guesser increases the time it takes to identify and normalise input files. To speed file identification, Xena attempts to determine the file type of the input file *before* passing it to every guesser:

1. Xena checks the file for a magic number and file extension.
2. If both a magic number *and* file extension are detected, Xena compares the input file values with a pre-existing set of magic number and file extension pairs.
3. If Xena finds a match, the file is passed to the corresponding guesser to confirm the file type.
4. If the guesser confirms the file type of the input file, the file is passed to the corresponding normaliser.
5. If any of the above checks are negative, the input file is passed to each guesser in-turn to establish a “best guess” of the file type (see Section 5.2).

The following diagram describes how Xena attempts to identify the file format of an input file in the first instance:

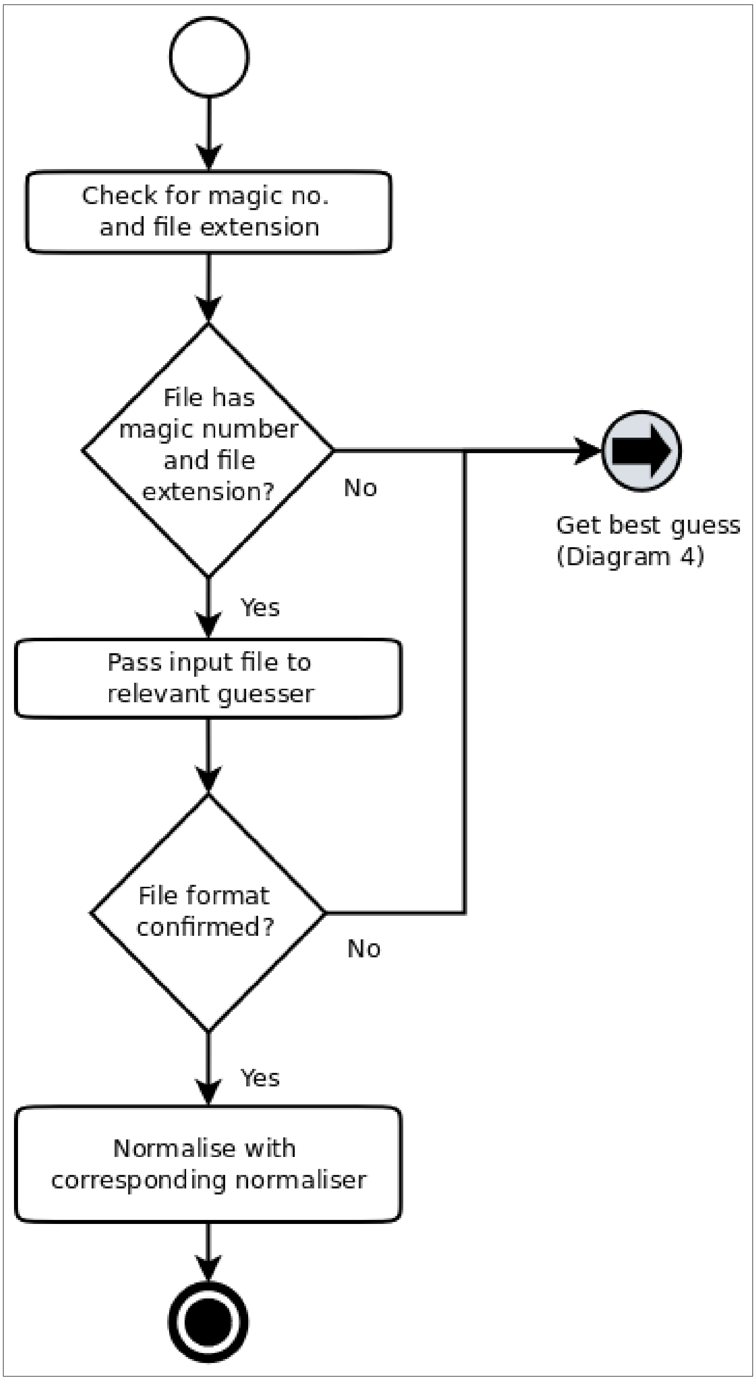
Check for magic no. and file extension
File has magic number and file extension?
No
Get best guess (Diagram 4)
Yes
Pass input file to relevant guesser
File format confirmed?
No
Yes
Normalise with corresponding normaliser

*Diagram 3: Initial identification attempt*

## *5.2 Best guess identification*

Xena performs a best guess file format identification if the initial identification attempt fails to determine the file format of the input file. The best guess identification process involves passing the input file to each guesser in turn. The input file is analysed by each guesser and a score is generated. The highest score is taken as the most probable indication of the file format.

1. The Plugin Manager passes the input file to each of the guessers in each loaded plugin.
2. Each guesser attempts to determine the likelihood of the input file being a particular format. The guesser compares attributes of the input file against the attributes of the guesser file type. Depending on the nature of the guesser, one or more of the following file attributes may form part of the comparison: MIME type; file extension; magic number; data in the body of the file (such as UTF-8 character encoding).
3. Each guesser returns a score which represents how closely the input file matched the attributes of the guesser file type (see Section 4 for more information).
4. A list of the scores from all guessers are ordered from highest to lowest.
5. The guesser with the highest score indicates the input file's most likely file type.
6. Based on the predicted file format, the corresponding normaliser is selected to normalise the input file.

The following diagram describes how Xena generates a best guess identification of file format:

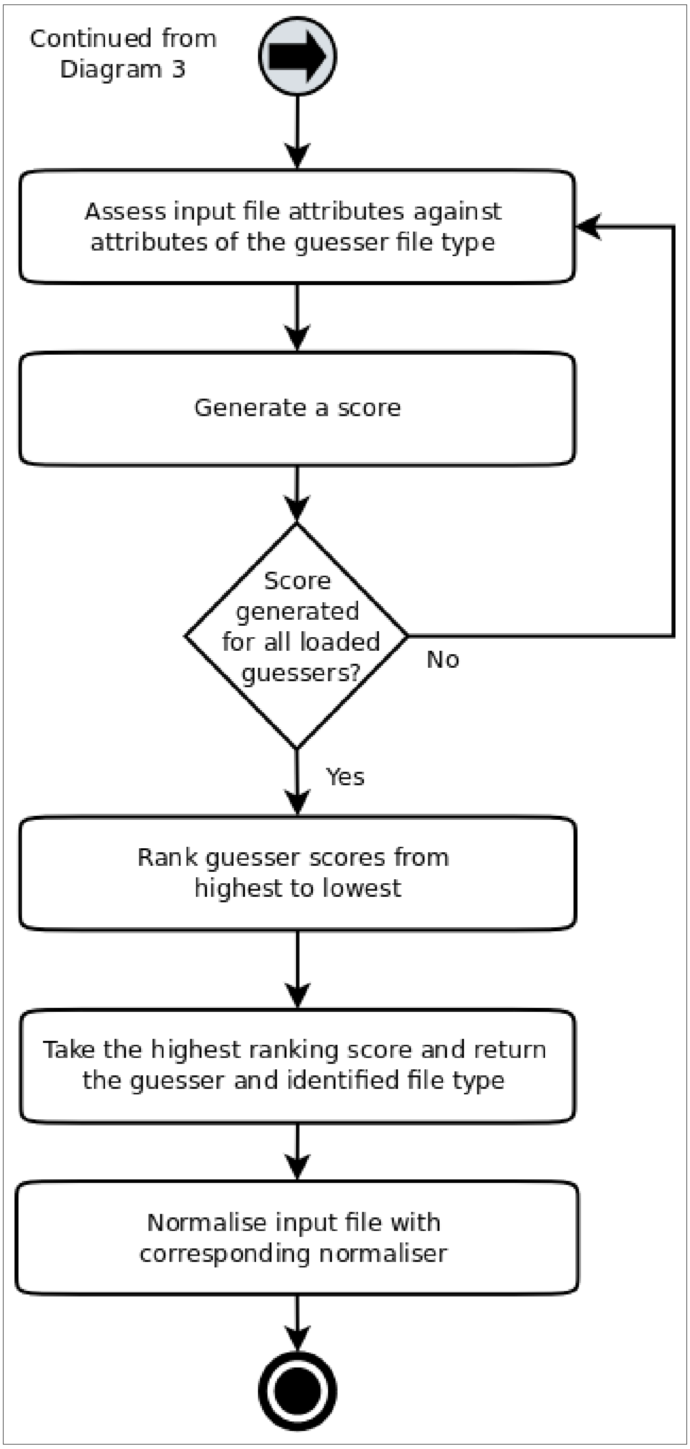
Continued from Diagram 3
Assess input file attributes against attributes of the guesser file type
Generate a score
Score generated for all loaded guessers?
No
Yes
Rank guesser scores from highest to lowest
Take the highest ranking score and return the guesser and identified file type
Normalise input file with corresponding normaliser

*Diagram 4: Best guess identification*

# 6 Appendix A – Xena plugins

The plugins in Xena version 5.0.0 are listed in the following table.

| [7]Plugin | Handles these types of files ... |
|---|---|
| archive | • Compressed archives (gzip, bzip2, war, zip)<br>• Uncompressed archives (jar, tar, zip) |
| audio | • Free Lossless Audio Codec (flac)<br>• Audio Interchange File Format (aiff)<br>• Broadcast Wave File (bwf)<br>• MPEG-2 audio layer 3 (mp3)<br>• Speex (spx)<br>• Vorbis (ogg, oga)<br>• Wave Audio File (wav) |
| csv | Text files consisting of comma-separated values |
| email | • Email (eml)<br>• Mailbox (mbx, mbox)<br>• Outlook Mail Message (msg)<br>• Outlook Personal Information Store (pst) |
| html | • Active Server Page (asp, aspx)<br>• HTML (htm, html)<br>• XHTML |
| image | • Portable Network Graphics (png)<br>• Bitmap (bmp, gif, pcx, pnm, ras, xbm)<br>• Photoshop (psd)<br>• Tagged Image File Format (tiff)<br>• Windows Cursor (cur)<br>• Open Document Graphics (odg)<br>• Joint Photographic Experts Group (jpeg)<br>• Scalable Vector Graphics (svg) |
| multipage | Multipage images (such as multipage tif or animated gif) |

[7] Video formats are currently not supported. Video files are normalised using the binary normaliser.

| Plugin | Handles these types of files ... |
|---|---|
| office | • Open Document Format (odf)<br>• Open Document XML (fodt)<br>• OpenOffice.org XML (stw, stc, std, sti, sxg, sxm)<br>• Excel (xls, xlsx, xlt)<br>• PowerPoint (pot, pps, ppt, pptx)<br>• Rich Text Format (rtf)<br>• Symbolic Link (slk)<br>• StarOffice (sdd, sdc, sdw, sxc, sxi, sxw)<br>• Word (doc, docx, dot)<br>• Windows Write Document (wri)<br>• Word Perfect (wpd) |
| pdf | Portable Document Format (pdf) |
| plaintext | Plain text in Unicode or ASCII |
| project | Project (mpp) |
| website | • ARC file format (arc)<br>• MIME HTML (mht)<br>• Web ARChive (warc) |
| xml | • Extensible Markup Language (xml)<br>• Extensible Stylesheet Language (xsl)<br>• XSL Transformations (xslt) |

# 7 Appendix B - Glossary

| Term | Definition |
|---|---|
| ASCII | American Standard Code for Information Interchange. A character encoding scheme for representing the English alphabet and punctuation (limited to 128 characters). Common character encoding format for plain text files (see also **Unicode**). |
| Base64 | Representation of binary data in an ASCII string format. |
| Binary | A system of counting using 1s and 0s. A binary file is a computer file which may contain any type of data for computer processing and storage. |
| Binary Normalisation | Base64 encoding of the content of a digital object, which is then wrapped in XML metadata. |
| Bit | A binary digit. In computing, a bit can either be a 1 or a 0. |
| BMP | Bitmap image file format. |
| BWF | Broadcast Wave Format. An extension of the Microsoft WAVE audio format. |
| Character Encoding | A system for representing individual characters with a code, such as a sequence of numbers. ASCII, ISO 8859 and Unicode are some popular character encoding schemes. |
| CSS | Cascading Style Sheets. A style sheet language used to describe the formatting of a document written in a markup language, such as **XML** or **HTML**. |
| CSV | Comma-separated values |
| DOC | Microsoft Word Document. |
| DOCX | Microsoft Office Open XML Document. |
| EPS | Encapsulated PostScript. A file format containing vector and sometimes bitmap data. |
| File | For the purposes of this document, a file is a digital file or computer file.<br>The term, **Digital Object** (from the Open Archival Information Systems Reference Model), has not been used in order to prevent confusion with the term **Object**. |
| FLAC | Free Lossless Audio Codec. A free and open source software tool and file format for lossless audio data compression. |
| GIF | Graphics Interchange Format. |
| GZIP | A software application used for file compression. |
| HTML | HyperText Markup Language. The main markup language for web pages. |
| JAR | Java Archive. A JAR file combines many other files into one. |

| Term | Definition |
|---|---|
| JPEG | Joint Photographic Experts Group file. A file format which employs a lossy compression for digital images. |
| Magic Number | A numeric or text value used to identify a file format. |
| Metadata | Data about other data. |
| MBX | Mailbox message file. A mailbox or mail folder that contains Microsoft Outlook Express e-mail messages. |
| MIME type | Multi-purpose Internet Mail Extensions. RFC2045 Internet standard allowing email to support attachments and non ASCII text.<br>Defines the kind of data formatting used by a particular file. |
| MP3 | MPEG-2 audio layer 3 (mp3) audio file. A lossy compressed audio format developed by the Moving Picture Experts Group. |
| MPP | Microsoft Project file. |
| Normalisation | Normalisation is the process of converting input files to an appropriate preservation file format. Conversion to the appropriate preservation file format depends on accurate detection of the input file format. |
| Normaliser | The normaliser is a component (Java object) of a Xena **plugin** responsible for taking an input file and transforming it into a Xena file. |
| Object | An object is a cohesive cluster of data and behaviour - an object contains information and can perform functions.[8]<br>In object-oriented programming terms, an object is an instance of a class. |
| ODF | Open Document Format. An XML-based file format for representing spreadsheet, text or presentation data. |
| ODG | Open Document Graphics file. |
| PDF | Portable Document Format. |
| Plugin | Plugins are a set of software components that add specific capabilities to a larger software application.<br>**Xena** plugins are one or more compiled Java classes that may be bundled in a Java Archive (JAR). To process digital records, Xena utilises plugins for various categories of file types. For example, audio, email and image. |
| PNG | Portable Network Graphics file. An image format that employs lossless data compression. |
| PSD | Adobe Photoshop document. An image file created by Adobe Photoshop. |
| PPT | Microsoft Powerpoint Presentation. |

[8] J. Arlow and I. Neustadt, UML 2 and the Unified Process 2$^{nd}$ Edition, Addison-Wesley, 2005

| Term | Definition |
|---|---|
| PPTX | Microsoft Powerpoint Office Open XML Presentation. |
| PST | Personal Storage Table. A Microsoft Outlook file format used to store email messages, contacts other data. |
| RTF | Rich Text Format file.  A method for encoding formatted text and graphics for transfer between applications. |
| SQL | Structured Query Language. A database computing language for managing the contents of a relational database. Includes insertion, query, update and deletion of data. |
| SVG | Scalable Vector Graphics. An XML-based file format for describing two-dimensional vector graphics. |
| TAR | Consolidated Unix File Archive. A file archive in an uncompressed format created by the Unix Tar utility. |
| TIFF | Tagged Image File Format. Graphics container that can store both raster and vector images. |
| Unicode | Industry standard for encoding text characters from most of the world's languages. It is a common character encoding format for plain text files. |
| UTF-8 | An 8-bit character encoding for **Unicode**, which is backwards compatible with the **ASCII** standard. |
| WAV | Waveform Audio file. |
| Xena | Digital preservation software developed by the National Archives of Australia. |
| Xena File | An XML file containing base64 encoded source file content, wrapped in metadata. |
| XHTML | eXtensible HyperText Markup Language. |
| XLS | Microsoft Excel Spreadsheet format. |
| XLSX | Microsoft Office Open XML Workbook. |
| XML | eXtensible Markup Language. |
| XSL | eXtensible Stylesheet Language. |
| XSLT | XSL Transformations. An XML language for transforming XML documents. |
| ZIP | A lossless compressed file archive format. |

# 8 Appendix C – File identification example

This example shows how three types of files are treated by three Xena guessers.

The files used in this example are described in the following table. Just to make things harder for our guessers, none of the files in this example have file extensions.

| File | MIME type | File extension | Magic No. | Data |
|---|---|---|---|---|
| Plaintext | N/A | N/A | File starts with ***EF BB BF*** | UTF-8 character set |
| HTML | **text/html** | N/A | Contains ***<html>*** tag | ASCII character encoding<br>Contains ***<html>*** tag in first 100 lines |
| Word Document | N/A | N/A | *File starts with* ***D0 CF 11 E0*** | ISO/IEC 8859-1 (Latin 1) character encoding |

*Attributes of three sample files*

The following sections show how each of the guessers generates a score for each of the three sample files (see Section 4 for the weightings used to calculate the various scores for each attribute); the highest positive score indicates the most likely file format.

## *8.1 HTML Guesser*

| File | MIME type score | File extension score | Magic No. score | Data score | Total score |
|---|---|---|---|---|---|
| Plaintext | 0 | 0 | -10000 | -30 | **-10030** |
| HTML | 60 | 0 | 50 | 30 | **140** |
| Word | 0 | 0 | -10000 | -30 | **-10030** |

*Scores given by the HTML guesser for three sample files*

## 8.2 Plaintext Guesser

| File | MIME type score | File extension score | Magic No. score | Data score | Total score |
|---|---|---|---|---|---|
| Plaintext | 0 | 0 | 50 | 30 | **80** |
| HTML | -10000 | 0 | 0 | 30 | **-9970** |
| Word | 0 | 0 | -10000 | 30 | **-9970** |

*Scores given by the Plaintext guesser for three sample files*

## 8.3 Word Guesser

| File | MIME type score | File extension score | Magic No. score | Data score | Total score |
|---|---|---|---|---|---|
| Plaintext | 0 | 0 | -10000 | 30 | **-9970** |
| HTML | -10000 | 0 | 0 | 30 | **-9970** |
| Word | 0 | 0 | 50 | 30 | **80** |

*Scores given by the Word guesser for three sample files*