Extensions of the UNIX File Command and Magic File for File Type Identification

William Underwood

Technical Report ITTL/CSITD 09-02

September 2009

Computer Science and Information Technology Division
Information Technology and Telecommunications Laboratory
Georgia Tech Research Institute
Georgia Institute of Technology

The Army Research Laboratory (ARL) and the National Archives and Records Administration (NARA) sponsor this research under Army Research Office Cooperative Agreement W911NF-06-2-0050. The findings in this paper should not be construed as an official ARL or NARA position unless so indicated by other authorized documentation.

ABSTRACT

File format identification is a core requirement for digital archives. The UNIX file command is among the most promising technologies for file type identification. This report describes extensions to the file command and magic file that enhance their utility for file format identification in archival systems.

A File Format Library (database) has been created to manage information about file formats. This information includes file format name, MIME type, PRONOM Universal Identifier and file signature tests. There is a one-to-one correspondence between file formats and file signature tests. Precedence relations between file signature tests are explicitly expressed in the database. Published specifications for file formats are also collected in the library and are used to determine file signatures for the formats. When specifications have not been published for a file format, samples for files in those formats have been collected and analyzed to determine possible file signatures. File signature tests have been created for more than 800 file formats. Sample files for more than 500 of the file formats in the library have been created or collected for testing of the file signatures. These examples are included in the library

The Library includes links to file format software resources that are needed in archival processing of digital records. These include: file viewers/players, archive extractors, file format converters, password recovery software and repairers for damaged files.

The File Format Library supports the creation of a magic file from the file signature tests in the Library. The GTRI File Type Identifier is a graphical user interface to the file command and the magic file created from the File Format Library. The file command and magic tests have been applied to examples of 500+ file formats from the File Format Library. These tests have led to refinement of the file signature tests and discovery of the precedence relationships among file signature tests.

The National Archives (TNA) of the UK provides a public registry of file format information (PRONOM). This information includes file signature patterns expressed as regular expressions. TNA also provides a tool (DROID) that uses these file signature patterns for file format identification. This approach to file type identification is also promising and seems to be primarily limited by the small number of file signature patterns in the PRONOM registry. GTRI is collaborating with TNA to enhance the content of the registry and the performance of the DROID file format identifier.

TABLE OF CONTENTS

1. Introduction

Automated file format identification is a necessary feature for the ingestion of digital objects into an archive. Such identification is needed to insure that the files received from a creator have the expected file formats so that the archive is able to preserve the files and make them available to the public. Knowledge of the file formats is necessary to insure that viewers/players are available for the files, for conversion of legacy file formats into standard, current or persistent object file formats, for extraction of files from archive files, and for repair of damaged files.

The file command and magic file available in the Linux and BSD flavors of UNIX is probably the most widely used tool for file format identification. The tests for identifying file formats in the magic file have been and remain the largest repository of information on file signatures in the world. However, the file command and magic file lack some features for file format identification that are required by digital archives.

The primary objective of the research described in this report is to identify the most promising technology for reliable file format identification and to advance this technology to meet the needs of the National Archives and Records Administration. The specific purpose of this report is to describe extensions made to the UNIX file command and magic file to improve the management of file format information and to increase the reliability of file type identification. These extensions include a File Format Library for managing file format information including file signature criteria that can be used to identify file formats. The library is also a repository for file format specifications and software for viewing/playing files, extracting files from archive files, recovering passwords, and repairing damaged files. It also contains sample files for the file formats in the library.

Section 2 of this report discusses the concepts of file types and file type identifiers. Section 3 briefly summarizes features of the file command and magic file and discusses some of their limitations. Section 4 describes a File Format Library that supports the management of file format information and that supports the creation of magic files used by the file command. Section 5 describes a graphical user interface for file type identification that is based on the file command and a magic file created from the File Format Library. Section 6 describes related research and development. Section 7 summarizes the results.

2. File Format Signatures and External File Identifiers

2.1 Basic Concepts

In the context of data storage and transmission, a *file* is a sequence of bits in which a data representation or computer instructions internal to a computer program has been encoded according to a file format so that it can be stored on a storage medium or transmitted over a network communication link. When the resulting file is read by a computer operating system, it is either decoded and executed by the computer, or passed to a computer program and decoded according to the file format to create a copy of the original internal data representation.

An *executable file* is a serialization of encoded computer instructions. A *script file* is a file that contains instructions for an interpreter or a virtual machine.

A *data file* is an external data representation in a sequence of bits of an internal data representation that can be stored on a storage medium or transmitted across a communication network. When the resulting file is reread according to the file format, it can be used to create a copy of the original internal data representation.

In object-oriented programming, *serialization* is the process of encoding an object into an architecture independent serial format for storage or transmission across a communication network. When the resulting series of bits is decoded according to the serialization format, it can be used to create a semantically identical copy of the original object. Such methods of serialization result in persistent objects that because of their architecture independence are not subject to obsolescence of computer platforms (hardware and operating systems). Examples of such formats include the Hierarchical Data Format (HDF), Comma Separated Values (CSV), and JavaScript Object Notation (JSON).

A *file type* (or *file format class*) is class of files with the same file format. A *file format signature* is invariant data in a file format that can be used to identify the file type (or format) of a file. In the UNIX operating system (including flavors such as BSD, Linux and Solaris), file signatures are referred to as *magic numbers*. In contrast to file format signatures, a *file signature* is a checksum or hash code of a file that can be used to check the integrity of the file

2.1 External Format Identifiers

A unique identifier is needed for file formats that is external to the file but can be linked to the file so that file signatures do not need to be checked every time a file is accessed. File name extensions and metadata stored in the operating system are two approaches that are used. MS-DOS and Windows file names use a file name extension to distinguish different file types. However, file extensions alone are often not enough to discriminate file types. For instance, file extensions such as DOC are ambiguous, since there are several applications that create files with that extension but have different file formats. Furthermore, there are WordPerfect document files that do not have the .DOC extension recommended by the WordPerfect manual. Instead, the document creator avails himself of the filename plus the filename extension to create a longer mnemonic filename. These extended names sometimes result in an extension used for another file type. For instance, SPEECH.COM is a user-created WordPerfect document file discovered in the Bush Presidential e-record collection that contains a speech to the Commonwealth Club. However, the .COM extension is also used to represent a MSDOS compressed executable file.

An alternative way of identifying file formats was developed by Apple Computer for the Macintosh OS Hierarchical File System. Each program installed has an associated *creator* code. This is a 3-4 letter code that tells the MacOS Finder which program created a file in the file system. Each time an application writes a file to the file system, a creator code as well as a file type code are stored as part of the directory entry for the file. The *file* type code is also a 3-4 letter code that tells the MacOS Finder the format of the file. The combination of creator and file

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.