# py360 User Guide

## Description

py360 is a python module for analysing Xbox 360 file systems and their associated proprietary file formats. In particular py360 provides a FUSE file system driver and file parsers for XTAF, XDBF, STFS and Account files from Xbox 360 hard drives. It is designed to aid forensic examination of the file system.

Components:

- report360.py - Client application that creates representations of various Xbox 360 files
- mount.py360 – Shell script wrapper for py360/py360.py
- py360/py360.py - FUSE filesystem driver
- py360/partition.py - Partition class for parsing XTAF files
- py360/stfs.py - STFS class for parsing STFS files
- py360/xdbf.py - XDBF class for parsing GPD/XDBF files
- py360/constants.py - Various classes containing constants such as region code mappings
- py360/account.py - Account class for decrypting/parsing Account files
- py360/xboxmagic.py - Class for determining the type of Xbox 360 related files
- py360/xboxtime.py - Functions for converting Xbox 360 time formats to unix time

## Installation

The installation of py360 is straight forward as long as the requirements are met. Largely all that is required is a Python interpreter and a Unix-like Operating System.

### Requirements

- Python 2.6
- Ubuntu 10.04LTS
- Minimum suggested hardware: Dual core x86 CPU, 1Gb RAM
- Python-fuse (for mount.py360 / py360.py)

The requirements represent the specifications of the development environment used to create py360 are in most instances are merely a known compatible configuration. Python 2.6 is the recommended Python interpreter and Python 3.0+ is explicitly not compatible. Any Unix-like operating system with Python 2.6 will work with py360 and Windows operating systems will work with minor code changes with the exception of the FUSE components (py360.py / mount.py360).

### Setup

If all the requirements are met there are no explicit installation that needs to take place. Optionally the py360 directory of the distribution could either be copied to the local python library path (for example /usr/lib/python2.6/dist-packages) or for the path to that directory added to the system python path to simplify the development of additional tools with the py360 module.

# Tools

## py360.py

py360.py contains the code to mount an XTAF partition (read only)

```
Usage: python py360/py360.py DEVICE MOUNTPOINT OPTIONS
```

DEVICE is the image file or block device that contains the XTAF partitions. If the device is a full disk image the data partition (starting at 0x130EB0000) is mounted, otherwise the partition that starts at offset 0 is used.

MOUNTPOINT is the directory to mount the file system on

OPTIONS are passed to FUSE, common options include: -d for debug , -o uid=UID to set the owner UID of the mounted files. See mount.py360 and the FUSE documentation for more options.

## mount.py360

mount.py360 is a shell script that will mount an XTAF partition using the most common options.

```
Usage: ./mount.py360 DEVICE MOUNTPOINT
```

DEVICE is the image file or block device that contains the XTAF partitions. If the device is a full disk image the data partition (starting at 0x130EB0000) is mounted, otherwise the partition that starts at offset 0 is used.

MOUNTPOINT is the directory to mount the file system on

## report360.py

report360.py contains code that provides detailed textual representations of Xbox 360 file structures. It also will parse a XTAF partition and recursively print information about all the structures and optionally write all the images files found to disk.

```
Usage: python report360.py DEVICE OUTPUT_DIRECTORY
```

DEVICE is the image file or block device that contains the XTAF partitions. If the device is a full disk image the data partition (starting at 0x130EB0000) is mounted, otherwise the partition that starts at offset 0 is used.

OUTPUT_DIRECTORY is the optional parameter that specifies where to write detected images to. If this parameter is missing no images will be written. Writing image to disk increases the run time of the tool by as much as a factor of 10.

The textual representations of the Xbox 360 file structures will be written to STDOUT while progress and processing information will be printed to STDERR. It is a recommended that STDOUT is redirected to a file as a typical XTAF file can create 10,000 lines of output. Run times range from about 1 minute for a simple XTAF files without image writing up to 15 minutes for larger images with image writing.

### xdbf.py

xdbf.py contains the classes that process XDBF files that contain information about users and games, these files are generally found inside STFS files. xdbf.py also will give a brief textual summary of XDBF files and write their embedded images to disk.

```
Usage: python py360/xdbf.py OPTIONS XDBF_FILE XDBF_FILE_2...
```

Option: `-p DIRECTORY` write embedded images to DIRECTORY

XDBF_FILE is the XDBF file to summarize or extract images from, it can be specified multiple times.

### stfs.py

stfs.py contains the classes that process STFS container objects that contain user profiles as well as games and media downloads. stfs.py can also be run to extract the contents of a STFS file to disk.

```
Usage: python py360/stfs.py INPUT_FILE OUTPUT_DIRECTORY
```

INPUT_FILE is the STFS file to process

OUTPUT_FILE is the directory to write the contained files to

## API

py360 is designed to be easy to use and easy to extend, it is comprised of a main class for each of the data types it supports and additional classes that represent record components inside each data type. The main classes can generally be instantiated with a file name to open or a python 'file-like' object and the record classes are generally instantiated by the main class. Almost all classes have a __str__ method and can be printed. Full class documentation is available in the doc directory of the source tree, the following is a brief overview of the main classes and their most important methods and members.

### py360.partition

Partition is the overarching class that processes a disk image. You provide it with the file name of the disk image or block device containing the XTAF partition and optionally set whether thread safety features are enabled (off by default for performance reasons). After initialisation the allfiles dict will have a fileobj or directory object for each file or directory on the image and the rootfile member will have a directory object representing the root directory.

Important components

- allfiles - Dictionary of FileObjs (and Directory objects), key is the full path string
- rootfile - Directory obj representing the root directory, it can be used as the root of a tree
- read_file - Takes either a filename or a fileobj via a named parameter and returns file data
- open_fd - Takes a file name and returns a XTAFFD object which provides read, seek and tell

## py360.stfs

STFS is the class that processes Xbox 360 Secure Transacted File System (STFS) containers. Its constructor requires either a file name or a 'file-like' object to process (an XTAFFD or StringIO object works correctly).

Important components

- allfiles - Dictionary of FileListig objects, key is the full path string
- read_file - Takes a FileListing object and returns the file (optional size and offset parameters)
- get_blockhash - Given a block number it returns the block hash object associated with it

## py360.account

Account is the class that decrypts and processes Xbox 360 account blocks found inside user profile STFS files (any file called 'Account' with the length of 404 inside an STFS file is likely an Account Block). Account blocks contain information about Xbox Live accounts as well as offline Xbox 360 accounts. The constructor requires a buffer containing the encrypted content (404 bytes long).

Important components

- decrypt - Decrypts the Account Block, primarily for internal use
- gamertag - Name of the account
- live_account - A boolean denoting whether this is a Live or offline Xbox account

## py360.xdbf

XDBF is the class that processes Xbox Database Files (XDBF) and in particular Gamer Profile Data files that are found within user profile STFS files with the extension gpd. Its constructor requires either a file name or a 'file-like' object to process (an XTAFFD or StringIO object works correctly). XDBF objects contain dictionaries of Achievement, Title and Setting objects as well as strings and buffers containing PNG files.

Important components

- achievements - Dictionary of Achievements, key is the id number
- titles - Dictionary of Titles, key is the id number
- settings - Dictionary of Settings, key is the id number (check py360.constants for mappings)
- images - Dictionary of strings containing PNG data, key is the id number
- strings - Dictionary of UTF-16-BE strings, key is the id number
- entries - List of Entry objects, each one corresponds to an achievement, title, etc.

### py360.xboxmagic

Xboxmagic is a module containing functions for determining the type of an Xbox 360 related file. It contains one function per type it recognises and the function and a function that tests 'file like' objects and data buffers.

Important components

- find_type - Given a 'file like' object or data buffer return a string describing its type

### py360.xboxtime

Xboxtime is a module containing functions for converting Xbox 360 time formats into Unix Epoch time. In particular it handles FAT time and Windows File Time.

Important components

- fat2unixtime - Provided FAT time and date components return seconds since 1/1/1970 00:00
- filetime2unixtime – Provided a 64 bit Windows File Time convert it to Unix Epoch.

### py360.constants

Constants is a module containing classes that provide mappings between constants and their values. Generally only large tables such as the region code table are in this module, other constants are defined in the class they are relevant to.

Important components

- STFSHashInfo.types - Dictionary mapping STFS block state bytes to their states
- GamerTagConstants.GamerZone - Maps Gamer Zone numbers to their names
- GamerTagConstants.Region - Region numbers to country name dictionary.
- GPDID.types – GPD ID codes to descriptive strings dictionary