Disk Image Preservation - Feature #10818

HFS disk image tasks [6 support tickets]

01/26/2017 12:00 PM - Nick Wilkinson

Status:	In progress	**Start date:**	
Priority:	Critical	**Due date:**	
Assignee:	Joel Dunham	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Sponsored:	No	**Requires documentation:**	
Pull Request:			

Description

Project requirements are being recorded here: https://wiki.archivematica.org/Improvements/Disk_Image_Preservation

History

#1 - 02/01/2017 12:28 PM - Nick Wilkinson

- Assignee set to Holly Becker

#2 - 02/01/2017 03:42 PM - Nick Wilkinson

- Description updated

#3 - 02/07/2017 03:31 PM - Holly Becker

- Status changed from New to In progress

For the sleuthkit upgrade:

I tried to upgrade sleuthkit from 4.1.3 to 4.4 and ran into dependency version issues.

- OS: Ubuntu 14.04.5 LTS
- SleuthKit Version: 4.1.3

Steps:

1. Install Sleuthkit
   - download sleuthkit-4.4.0 from https://www.sleuthkit.org/sleuthkit/download.php
   - tar xzfv sleuthkit-4.4.0.tar.gz
   - ./configure
   - installing without libewf and afflib support
   - make
   - sudo make install
2. Check upgrade
   - mmls -V outputs The Sleuth Kit ver 4.4.0
   - fiwalk --help outputs fiwalk: error while loading shared libraries: libtsk.so.13: cannot open shared object file: No such file or directory
   - /usr/local/lib/libtsk.so.13 exists and is a symlink to /usr/local/lib/libtsk.so.13.2.0 which also exists
3. Uninstall upgrade sleuthkit
   - sudo make uninstall
4. Troubleshoot
   - dpkg -s sleuthkit reveals a dependency named libtsk10
   - Try installing with sudo apt-get install libtsk13
   - libtsk13 doesn't exist - only available on 16.04 and later http://packages.ubuntu.com/search?keywords=libtsk13

This will have to be looked into further to decide whether to install libtsk13 on older Ubuntu versions, or install an older sleuthkit, or support newer versions of Ubuntu (16.04+)

#4 - 02/16/2017 02:36 PM - Holly Becker

I did some analysis of the tools available to work with raw disk images and HFS in particular. From that, I created several sample FPR entries to deal with them. All work done on the sample uclalsc_ml_227_026.img

Identification

None of the tools currently in use can correctly identify the format of the .img file with an HFS filesystem.

- Fido & Siegfried return x-fmt/159 (GEM Image), fmt/383 (VICAR (Video Image Communication and Retrieval) Planetary File Format) and fmt/625,(Apple Disk Copy Image) based on the file extension
- fmt/625 might be the correct identification
- 'file' returns "PC formatted floppy with no filesystem". However, 'file' on Ubuntu 14.04 cannot identifyr the E01 image, and just returns "data"
- blkid returns LABEL="Chris disk" TYPE="hfs"
- hfs2dfxml [1] generates XML output with information about the filesystem and contained files.

Ideally, we would find the magic bytes that blkid and file are using to identify the format and submit it to PRONOM. However, that may not be possible, and PRONOM may not be interested in the format beyond raw disk image, not including the filesystem.

FPR changes

The workaround solution is to create an FPR entry for a raw disk image with HFS, and modify identify by extension to recognize it.

- New FormatVersion: Format: Raw Disk Image; Description: Raw Disk Image (HFS filesystem)
- Modify Identify by File Extension like [2]
  - Command:

```
from __future__ import print_function
import os.path
import subprocess
import sys

def file_tool(path):
    return subprocess.check_output(['file', path]).strip()

def blkid(path):
    try:
        return subprocess.check_output(['blkid', '-o', 'full', path])
    except Exception:
        return ''

(_, extension) = os.path.splitext(sys.argv[1])

if extension:
    print(extension, end='')
    if extension in ('.img,'):
        output = blkid(sys.argv[1])
        if 'TYPE="hfs"' in output:
            print(' (hfs)')
else:
    # Plaintext files frequently have no extension, but are common to identify.
    # file is pretty smart at figuring these out.
    file_output = file_tool(sys.argv[1])
    if 'text' in file_output:
        print('.txt')
```

- New ID Rule: Format: Raw Disk Image (HFS filesystem); Command: Identify by File Extension; Output: ".img (hfs)"
- Disable Rule with output ".img", or modify to identify as Raw Disk Image

[1] https://github.com/cul-it/hfs2dfxml
[2] https://github.com/artefactual/archivematica-fpr-tools/blob/dev/issue-10818-hfs-disk-image/id/file-by-extension.py

Extraction

- Fiwalk does not recognize the filesystem, and cannot extract from it.
- hfsutils provides the hmount and hcopy commands, but hcopy is not recursive
- tsk_recover cannot recognize the filesystem, outputting "Cannot determine file system type (Sector offset: 0)Files Recovered: 0"
- hfsexplorer [3] provides a command line extraction tool for HFS filesystems. However, hfsexplorer is not packaged for Ubuntu, and must be installed manually.

To install hfsexplorer, download and extract it. By default it uses a GUI, but a command line interface is accessible from the hfsx.sh script. The script we want is unhfs.sh, which extracts files from the image.

FPR changes

To handle extraction, use hfsexplorer's unhfs command to extract all files from the hfs partition.

- Set up file identification FPR changes
- Install hfsexplorer somewhere Archivematica can run it from
- New FPR Tool: Description: hfsexplorer; Version: 0.23.1
- New Extraction Command:
  - Tool: hfsexplorer
  - Description: unhfs
  - Script Type: bash
  - Command:

    ```
    mkdir "%outputDirectory%"
    /home/users/hbecker/bin/hfsexplorer/bin/unhfs.sh -v -o "%outputDirectory%" "%inputFile%"
    ```

  - Output location: outputDirectory
  - Command Usage: Extraction
- New Extraction Rule: Purpose: Extract; Format: Raw Disk Image (HFS filesystem); Command: unhfs

[3] http://www.catacombae.org/hfsexplorer/

Characterization

For characterization, the hfs2dfxml [4] provides a nice XML output with metadata about the image. However, it is also not packaged for Ubuntu. To install it, follow the instructions in the README [5] by installing hfsutils & python-magic, cloning the repository and cloning the dependency dfxml in the correct location inside the repository.

However, it can only be run from inside the repository without a patch. Either clone my fork [6] and change branches, or apply the patch [7] yourself.

FPR changes

- Set up file identification FPR changes
- Install & patch hfs2dfxml somewhere Archivematica can run it from
- Disable "Delete packages after extraction"
- New FPR Tool: Description: hfs2dfxml; Version: git commit hash
- New Characterization Command:
  - Tool: hfs2dfxml
  - Description: hfs2dfxml characterization
  - Script Type: bash
  - Command:

    ```
    output=/tmp/temp_`uuid -v4`
    echo $(id)
    python /home/users/hbecker/bin/hfs2dfxml/hfs2dfxml/hfs2dfxml.py "%fileFullName%" $output
    cat $output
    rm $output
    ```

  - Output Format: Text (Markup): XML: XML
  - Command Usage: Characterization
- New Characterization Rule: Purpose: Characterization; Format: Raw Disk Image (HFS filesystem); Command: hfs2dfxml

However, that setup currently generates an error when run through Archivematica. "_call_hmount error: Failed to initialize HFS working directories: Permission denied" hfs2dfxml is being run, but generates an error when trying to call hfsutils. This requires further investigation.

[4] https://github.com/cul-it/hfs2dfxml
[5] https://github.com/cul-it/hfs2dfxml/blob/master/README.md
[6] https://github.com/Hwesta/hfs2dfxml/tree/patch-1
[7] https://github.com/cul-it/hfs2dfxml/pull/7/files

#5 - 02/28/2017 10:00 AM - Nick Wilkinson

- Status changed from In progress to Feedback

- Assignee changed from Holly Becker to Sarah Romkey

Hey Sarah, not sure if you've spoken with Holly about this -- she said she's blocked on it would be good for the client to do some testing based on the partial work she's done. Assigning to you for discussion with the client.

#6 - 03/01/2017 02:58 PM - Sarah Romkey

I'll look over this in the next day or two but in the meantime just a note that I added more disk image samples from Susan to our private sample data folder.

#7 - 03/03/2017 05:23 PM - Holly Becker

Deployment notes:

To test the additional FPR commands, two additional tools must be installed: hfsexplorer and hfs2dfxml. Neither have packages, and are currently installed from git.

hfsexplorer:

1. Go to http://www.catacombae.org/hfsexplorer/
2. Download the ZIP at link: Download application as ZIP file (cross-platform)
3. Extract ZIP into directory accessible by Archivematica (suggested: extract into /home/vagrant/hfsexplorer)
4. Record full path to 'hfsexplorer/bin/unhfs.sh' for use in FPR

hfs2dfxml

1. Install dependencies: sudo apt-get install hfsutils python-magic
2. Go to where the program should be installed (suggested: /home/vagrant)
3. Clone my patched version: git clone https://github.com/Hwesta/hfs2dfxml
4. Go into repo: cd hfs2dfxml/hfs2dfxml
5. Switch to branch with patch: git checkout -t origin/patch-1
6. Clone dependency: git clone https://github.com/simsong/dfxml/
7. Record full path to 'hfs2dfxml/hfs2dfxml/hfs2dfxml.py' for use in FPR

#8 - 03/04/2017 07:55 AM - Sarah Romkey

- Status changed from Feedback to Deploy

- Assignee changed from Sarah Romkey to Nick Wilkinson

Nick, can you assign to someone to deploy for internal and client testing? Holly summarized some notes, see above.

#9 - 03/06/2017 04:08 PM - Nick Wilkinson

- Assignee changed from Nick Wilkinson to Hector Akamine

Hi Hector, assigning to you for deploy -- see Sarah's note # 8. https://trello.com/c/OWtykPxH

#10 - 03/09/2017 04:53 PM - Hector Akamine

Re sleuthkit,

- Even Ubuntu 16.04 does not provide the latest version 4.4, only 4.2
- It looks like Misty had been building packages before (ref. https://internal.artefactual.com/gitweb/?p=packaging-sleuthkit.git;a=summary). There is also this: https://launchpad.net/ubuntu/+source/sleuthkit/4.4.0-2 . Maybe we can use these to build new packages. Investigating.

#11 - 03/21/2017 10:06 AM - Hector Akamine

- Assignee changed from Hector Akamine to Santiago Collazo

#12 - 03/24/2017 10:43 AM - Santiago Collazo

Hostname is am16hfs.archivematica.org , and user/pass is in lastpass

There are also two sftp accounts, ucla and nypl

#13 - 04/18/2017 04:52 PM - Nick Wilkinson

- Status changed from Deploy to Feedback

- Assignee changed from Santiago Collazo to Sarah Romkey

Hi Sarah, assigning to you for consideration.

#14 - 04/26/2017 09:33 AM - Nick Wilkinson

- Status changed from Feedback to In progress

- Assignee changed from Sarah Romkey to Joel Dunham

Hi Joel, assigning this back to you. Kelly has more details if you'd like to discuss next steps.

#15 - 06/06/2017 02:30 PM - Joel Dunham

The HFS disk image extraction and characterization functionality should work on
http://am16hfs.archivematica.org/transfer/ and it should also work on a new
system that is running Archivematica at branch dev/issue-10818-hfs-disk-images.

- Am branch: https://github.com/artefactual/archivematica/tree/dev/issue-10818-hfs-disk-images
- AM-fpr-admin branch: https://github.com/artefactual/archivematica-fpr-admin/tree/dev/issue-10818-hfs-disk-images

What has been Done

1. Added a new characterization command called "Fiwalk fallback to hfs2dfxml" which attempts to use fiwalk for characterizing disk images and, if that fails, attempts to use hfs2dfxml. Also set the "Output File Format" to "XML".
2. Created a new extraction command for disk images called "tsk_recover fallback unhfs" which attempts to extract a disk image using tsk_recover and then falls back to using unhfs if that fails. Set output file format to JSON so that the output can alter the default tool from "Sleuthkit" to "hfsexplorer" when appropriate.
3. Modified the Siegfried fpr_idcommand so that it attempts to run blkid when it would otherwise return 'UNKNOWN'; if blkid indicates that the file is an HFS disk image, then the fpr_idcommand returns a custom pronom id: "archivematica-fmt/6"
4. Modified MCPClient/clientScripts/extractContents.py so that the tool used in extraction will be listed in the eventDetail. In the HFS extraction case, this will be "hfsexplorer".

TODOs

- IMPORTANT: Investigate failure at Ingest > Verify checksums > Verify checksums generated on ingest: “Checksums do not match”, which occurs consistently with one particular .001 disk image transfer: M1126-0001.001. I tested with several other .001 and .img transfers and none of them exhibited the same behaviour. It turns out that hfs2dfxml.py modifies https://github.com/cul-it/hfs2dfxml/issues/12 the bitstream of the disk image in some cases. Next step is to try the hfsutils commands directly and see if the checksum changes in that case.

- Ensure that the demo server at am16hfs.archivematica.org has the updated code, including the acceptance tests (ensuring they pass).
- Set HOME env var in MCPClient/lib/archivematicaClient.py#L180-L183 (and ultimately in upstart), NOT in the command itself, i.e., not in fpr/migrations-data/fiwalk_fallback_hfs2dfxml.py
- Move the autoslug version change to a separate commit. I updated Django autoslug to the most recent version because vagrant provision was triggering an error related to a change since autoslug 1.7.1:
- Create ansible tasks to install the tools and their dependencies: unhfs, HFSutils, hfs2dfxml. Note: I have written a shell script (hfs-deps.sh) that does this already and which should be put in a GitHub gist.
- Condense the information in objectCharacteristicsExtension so that it doesn't have the FULL DFXML in there. Crucial information might be: How many files are in the disk image? How big is it? What are the date ranges, i.e., earliest file creation date and latest file modification date. - EAD ...
- Modify hfs2dfxml for optimizations: hcopy copies and deletes files, which is inefficient but easy to turn off.
- Test hfs2dfxml on filenames with weird characters; it is known to fail in some cases. (unhfs is a potential fallback here)
- Assemble more disk image samples for testing and figure out how to make those we have available for the acceptance tests (without being in a public repo).

Installation instructions

To install such a system using Archivematica's deploy-pub Vagrant repo, modify the vars-singlenode-1.6.yml file so that archivematica_src_am_version is set to "dev/issue-10818-hfs-disk-images." and provision: vagrant provision

In order to get the new HFS disk image functionality to work in Archivematica, the following tools and their dependencies must be manually installed:

- hfs2dfxml
- hfsexplorer

The following instructions were used on Ubuntu 14.04. Jump to the bottom for a shell script that can install all of these dependencies on a default Archivemcatica vagrant/ansible deploy.

To install hfsutils:

```
$ sudo apt-get update
$ sudo apt-get install hfsutils
```

Install python-magic. WARNING: it's important that you do NOT install this python-magic: https://github.com/ahupp/python-magic. Instead, you must install the python-magic (see https://github.com/threatstack/libmagic) that Ubuntu installs when you call the following:

```
$ sudo apt-get install python-magic
```

If your install has separate virtual environments for each Archivematica component, then the MCPClient virtualenv needs to have magic installed also:

```
$ sudo ln -s /usr/lib/python2.7/dist-packages/magic.py /usr/share/python/archivematica-mcp-client/lib/python2.7/site-packages/magic.py
```

Install (Holly Becker's patch of) the hfs2dfxml source in your home directory on the machine where Archivematica is installed:

```
$ pwd
/home/vagrant
$ mkdir bin
$ cd bin
$ git clone https://github.com/Hwesta/hfs2dfxml
$ cd hfs2dfxml/hfs2dfxml
$ git checkout -t origin/patch-1
$ git clone https://github.com/simsong/dfxml/
```

Install hfsexplorer (instructions repeated from above):

1. Go to http://www.catacombae.org/hfsexplorer/

2. Download the ZIP at link: Download application as ZIP file (cross-platform)
3. Extract ZIP into directory accessible by Archivematica (suggested: extract into /home/vagrant/hfsexplorer)
4. Make sure that the hfsexplorer directory is in the directory that extraction command expects it to be, i.e., /usr/local/hfsexplorer/bin/unhfs.sh

All of the above dependencies can be installed on a default AM deploy-pub vagrant install (Ubuntu 14) using the following bash script.

```
#! /usr/bin/env bash

# Install dependencies for HFS disk image processing in Archivematica on Ubuntu
# 14.04.
# Note: this is just for development purposes: these steps should be
# implemented in a platform-independent way using Ansbible in a future
# modification.

# hfsutils and python-magic
sudo apt-get update;
sudo apt-get install -y hfsutils python-magic;
sudo ln -s /usr/lib/python2.7/dist-packages/magic.py /usr/share/python/archivematica-mcp-client/lib/python2.7/site-packages/magic.py;

# hfs2dfxml
cd /home/vagrant;
mkdir -p bin;
cd bin;
git clone https://github.com/Hwesta/hfs2dfxml;
cd hfs2dfxml/hfs2dfxml;
git checkout -t origin/patch-1;
git clone https://github.com/simsong/dfxml/;

# hfsexplorer (unhfs.sh)
cd /home/vagrant;
mkdir -p downloads;
cd downloads;
wget -O hfsexplorer-0.23.1-bin.zip "https://downloads.sourceforge.net/project/catacombae/HFSExplorer/0.23.1/hfsexplorer-0.23.1-bin.zip?r=http%3A%2F%2Fwww.catacombae.org%2Fhfsexplorer%2F&ts=1494962350&use_mirror=cytranet";
mkdir -p /home/vagrant/hfsexplorer;
unzip -o hfsexplorer-0.23.1-bin.zip -d /home/vagrant/hfsexplorer;
sudo cp -r /home/vagrant/hfsexplorer /usr/local/;
```

Also make sure to modify the default processing config so that Siegfried is the file identification command during transfer and packages are not deleted after extraction: this allows the disk image to be characterized after its contents have been extracted.

#16 - 10/05/2017 10:27 AM - Joel Dunham

The following branches were rebased against the relevant qa branches and confirmed to work as described in the above comment.

- Am branch: https://github.com/artefactual/archivematica/tree/dev/issue-10818-hfs-disk-images
- AM-fpr-admin branch: https://github.com/artefactual/archivematica-fpr-admin/tree/dev/issue-10818-hfs-disk-images

#17 - 12/11/2017 03:21 PM - Justin Simpson

HFS is now included in PRONOM v93 as fmt/1105. Steps left to get support for working with HFS disk images into a future Archivematica release

- Include siegfried v1.7.8 or later
- Produce a new FIDO release with PRONOM v93
- Update FPR-admin to PRONOM v93
- add new Characterize command (refactor from https://github.com/artefactual/archivematica-fpr-admin/commit/56f79dbc08271bb69dc22547487b797df445ff99)
- add new Extract command (refactor from same commit)
- add hfs2dfxml as new mcp client tool
- add hfsexplorer as new mcp client tool
- confirm if unhfs.sh is included in hfsexplorer or needs to be added separately
- update extractContents.py client script (refactor https://github.com/artefactual/archivematica/commit/ee07f37c5f5d5f3d820678d97f7d29447761550a)