Bulk Forensic Image Processor- V6.0

Release Notes & Guide

Release Notes V6.0

Major Updates:

- Added native support for processing mobile filesystem extraction ZIP archives.
- Added Add Single Source for directly queueing an individual VICS JSON, supported forensic image, or mobile extraction ZIP.
- Updated Breakpoint Processing Engine settings with separate forensic-image and mobile-ZIP carving options.
- Updated API/CLI mode to support mobile ZIP processing and current Breakpoint Processing Engine carving options.
- Updated look, with numerous UI/UX enhancements.

Mobile Extraction Support:

- Mobile ZIP processing can extract selected standalone files and create VICS-compatible JSON packages.
- Mobile processing preserves original file paths, filenames, timestamps, MD5 hashes, and file sizes where available.
- Added recovery of media from files with missing, unusual, or misleading extensions using file signature detection.
- Added embedded media recovery from common mobile storage locations including SQLite databases, cache files, previews, attachments, and related app storage.
- Added optional mobile recovery of embedded media from Office documents and embedded images from PDFs.

Breakpoint Processing Engine:

- Added optional recovery of embedded media from Office documents and embedded images from PDFs in forensic-image processing.
- Improved embedded SQLite and ThumbCache output metadata.
- Embedded recoveries now include MD5 and file size when available while maintaining VICS-compatible output.
- Updated Photorec libraries to 7.2

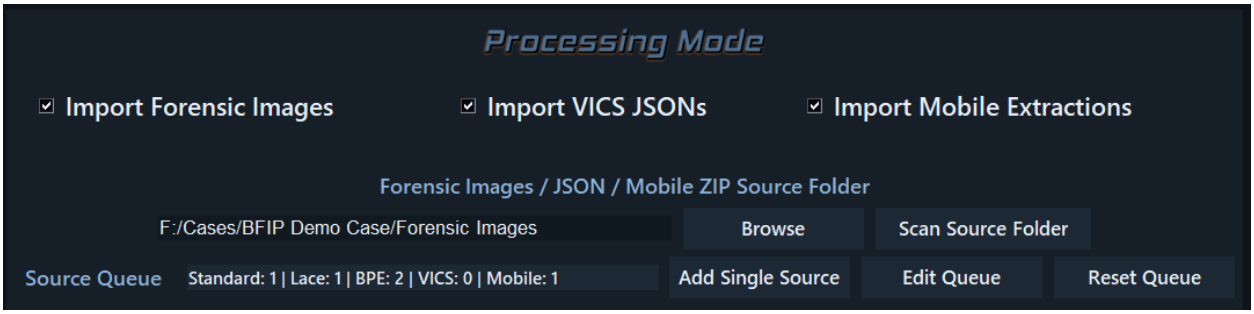
Requirements:

- Windows 10 or 11
- Griffeye DI or Processing Engine 24.3 or Newer
    - Pro/Advanced License *required* for Lace and Auto Case Creation and Import Functions.
    - Griffeye Plugins such as EXIF AI, Thorn, Brain, etc. must be installed and configured per instructions in Griffeye Analyze Forensic Market prior to use.
    - Pro/Advanced *not required* for *Break Point Processing Engine* Carving
- ***Griffeye Collaboration Server Integration requires:***
  Magnet Griffeye Connect CLI
  Version:24.4.1.0 or newer
  Newest Download available on Magnetforensics.com support page on Griffeye Operations/Enterprise Downloads section or here:

Contents

Processing Mode

BFIP offers multiple processing engines and source workflows depending on Griffeye license, examiner preference, and source type. The Advanced Source Queue Setup is used for configuring processing modes, adding sources from different paths, and mixing forensic images, VICS JSON packages, and mobile extraction ZIPs in a single run. Please see the Advanced Source Queue Setup section for details on how to add and configure sources.



Source Type

Import Forensic Images

- Select this Checkbox to have BFIP search for forensic images to add to import queue and process them based on configured Processing Engine Selection.

Import VICS JSON

- Select this Checkbox to have BFIP search for JSON Packages to add to import queue. Can be run alongside Forensic Image Import and/or independently.

Import Mobile Extractions

- Select this Checkbox to have BFIP search for supported mobile filesystem extraction ZIP archives. Mobile ZIPs are processed with the Breakpoint Mobile Processing Engine and output VICS-compatible JSON packages containing extracted files and metadata.

Scan Source Folder:

Select the parent folder where your case forensic images, JSON packages, and/or mobile extraction ZIP archives are stored. BFIP will search the specified folder and subfolders for supported sources and batch add them to the processing queue in one step. Source IDs are initially generated from the source filename with the extension stripped, but may be customized in the advanced source queue window.

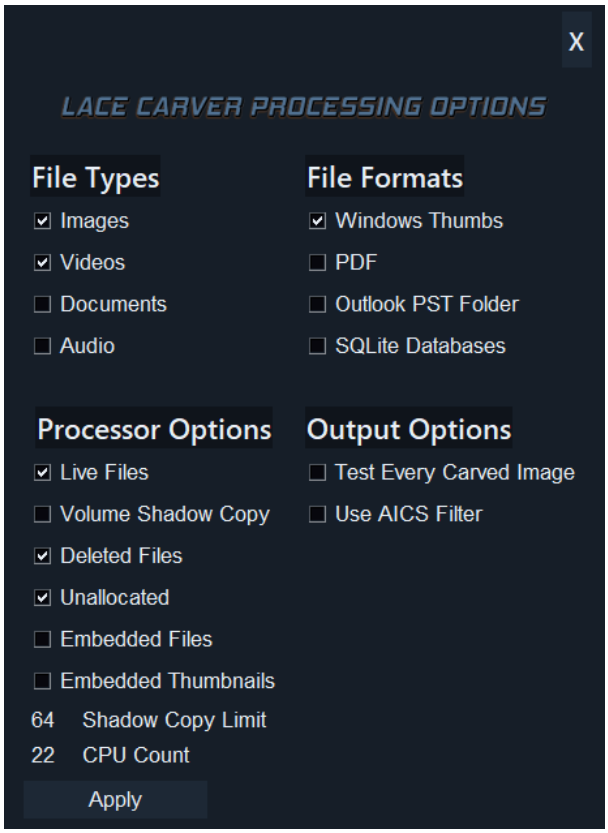
Processing Mode Engines

Standard

- Standard calls the standard/default import engine included with Griffeye.
- Includes Active Files and 'Flagged Deleted' Files.
- *Does not* recover data from unallocated space.
- *Supported Image Types: ['.001', '.bin', '.dd', '.dmg', '.e01', '.ewf', '.iso', '.raw', 'vhd']*

Lace

- Requires Lace Carver Addon with your Griffeye Adv. or Griffeye Processing Engine License.
- Completely replaces the Standard Griffeye Import Engine.
- Many Selectable Options to includes Active, Deleted, Unallocated, VSS, and Embedded Files
- Options Configurable via 'Lace Carver Options' menu.
- *Supported Image Types: ['.001', '.aa', '.aff', '.ad1', '.bin', '.dd', '.dmg', '.e01', '.ex01', '.ewf', '.iso', '.l01', '.raw', '.s01', '.smart', '.vhd', '.vmdk']*



Breakpoint Processing Engine

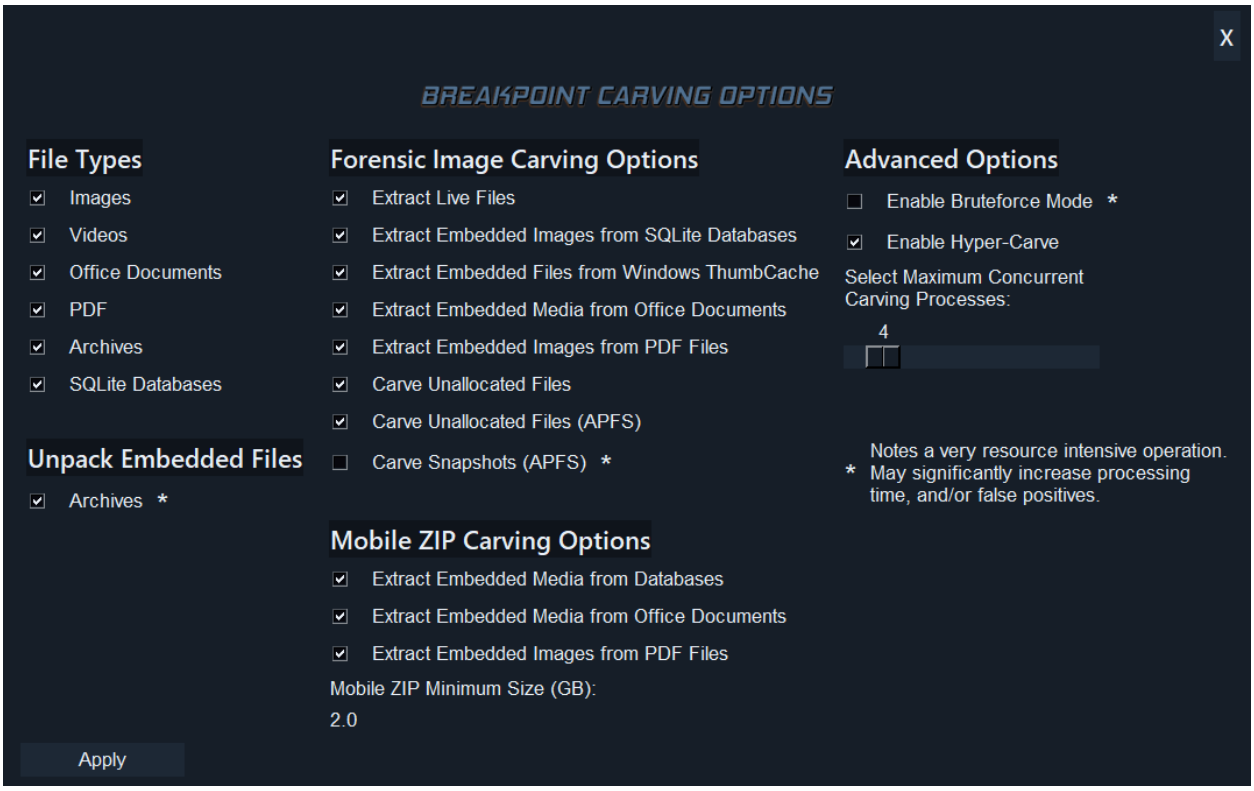
- Custom Processing Engine that provides a Hybridized import process leveraging several custom modules in combination with a fully automated implementation of PhotoRec.
- Includes additional support for parsing forensic images containing common file systems including but not limited to NTFS, FAT, APFS(Apple File System), with automated extraction of media files and import into Griffeye with *no additional addon plugins required.*
- Includes Active Files, Deleted Files, and Carving of Unallocated Files, Carving Files from APFS Snapshots, and Carved Archive Extraction.
- No additional licensing addons required.
- Ability to conduct several parallel carve processes using 'Hyper-Carve' option for significant reduction in typical carving time.
- Passes recovered data to custom VICS JSON generator to build out JSON containing notable metadata and fields (i.e. Physical Location, Files Paths, Deleted Status, Unallocated Status, etc.)
- *Supported Image Types: ['.001', '.bin', '.dd', '.dmg', '.e01', '.ewf', '.iso', '.raw', 'vhd']*

Breakpoint Mobile Processing Engine

- Processes Full Filesystem mobile extraction ZIP archives commonly generated by mobile forensic tools such as Graykey or Cellebrite.
- Extracts selected standalone files directly from the ZIP based on configured BPE file-type options, including images, videos, documents, PDFs, archives, and databases. This baseline extraction also includes targeted recovery from selected cache files, previews, attachments, and related mobile app storage.
- Preserves original mobile extraction paths, filenames, timestamps, MD5 hashes, and file sizes where available.
- Creates a VICS-compliant JSON package for import into Griffeye or other supported tools.
- When enabled, additional recovery options can identify and extract embedded media from databases, Office documents, and PDFs.
- Supported Source Type: Mobile filesystem extraction ZIP archives.

-

Breakpoint Processing Engine Options



File Types:

Images: Recover common image formats including JPG, PNG, GIF, BMP, TIFF, HEIC/HEIF, WebP, RAW camera formats, and related image types.

Videos: Recover common video formats including MP4, MOV, 3GP, AVI, MKV, WEBM, MPEG/MPG, MTS/M2TS, WMV, and related video types.

Office Documents: Recover common Office/OpenDocument files including DOC/DOCX, XLS/XLSX, PPT/PPTX, ODT/ODS/ODP.

PDF: Recover standard PDF files.

Archives: Recover common archive/container formats including ZIP, RAR, 7Z, ISO, and DMG.

SQLite Databases: Recover SQLite database files. (Mobile ZIP database embedded-media extraction is controlled separately under Mobile ZIP Carving Options.)

Unpack Embedded Files:

Archives: Processes RAR, 7Zip, ZIP, and DMG files recovered from unallocated carve and extracts embedded files for inclusion in the Griffeye import queue.

*Can significantly increase processing time and case size depending on the number and size of archives found.

Forensic Image Carving Options:

Extract Live Files: Have Breakpoint Processing Engine extract live and flagged deleted files from supported file systems.

Extract Embedded Media from SQLite Databases: Attempt to identify and extract media embedded in SQLite databases recovered from allocated or unallocated files.

Extract Embedded Images from Windows ThumbCache: Attempt to identify and extract images embedded in Windows ThumbCache databases.

Extract Embedded Media from Office Documents: Attempt to recover embedded image and video files from supported Office/OpenDocument containers.

Extract Embedded Images from PDF Files: Attempt to recover directly embedded image streams from PDF files. BFIP does not render or convert PDF pages.

Carve Unallocated Files: Have Breakpoint Processing Engine carve and extract selected file types from unallocated space.

Carve Unallocated Files (APFS): Force Breakpoint Processing Engine to conduct deep carve and extraction of selected file types from APFS volumes.

Carve Snapshots (APFS): Have Breakpoint Processing Engine check for APFS snapshots and recover deltas of files not located in the current live file system.

Mobile ZIP Carving Options:

Extract Embedded Media from Databases: Attempt to identify and extract selected embedded media from databases and related database storage inside mobile ZIPs.

Extract Embedded Media from Office Documents: Attempt to recover selected embedded media from Office/OpenDocument files found inside mobile ZIPs.

Extract Embedded Images from PDF Files: Attempt to recover directly embedded image streams from PDF files found inside mobile ZIPs.

Mobile ZIP Minimum Size (GB): Sets the minimum ZIP archive size used during bulk source-folder scanning. ZIPs selected manually with Add Single Source bypass this filter.

Advanced Options:

Bruteforce Mode: Enables PhotoRec brute force mode that can increase the number of fragmented files recovered.

*Can significantly increase processing time, false positives, and CPU demand.

Hyper-Carve: When enabled, Breakpoint Processing Engine initializes carving, extraction, and JSON creation for each forensic image in separate concurrent threads. Mobile ZIPs may run alongside forensic-image processing, but ZIP processing is kept to one ZIP at a time for stability.

Case Setup

Case Name and Location:

Create New Case

Enter the case name/# and storage location where you want your Griffeye Case Folder to be created. BFIP will automatically generate all needed additional case-folders and files based on the information you enter.

If you choose to use the 'Carve Only' option, the carved data will be output to the same path specified here.

Example: [Storage Path] + [Case#] = D:/Griffeye Cases/2024-123456/

Add Additional Sources to Existing Case

BFIP can also be used to add additional sources/data to an existing Griffeye Case. Select the radial for 'Add to Existing Case, and the locate the existing Griffeye ANCF case file using the

selection button. BFIP will automatically *add* any new sources/data to the existing case. Your existing case *WILL NOT* be overwritten.

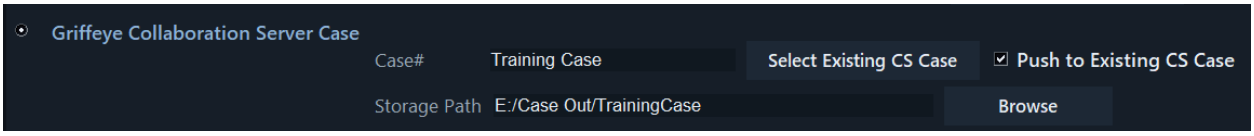
Griffeye Operations/Enterprise Collaboration Server(CS) Case Setup

In order to utilize Griffeye CS Integration, please ensure latest Griffeye Connect CLI is installed, and a connection to your Collaboration Server has been successfully completed in the BFIP 'Advanced Settings' menu.

Initial Configuration Settings Covered in Advanced Settings Menu Discussed Here.

Once Griffeye CS integration has been successfully enabled, users may conduct processing of forensic images using the Breakpoint Processing Engine, and recovered files can then be automatically pushed to cases on the configured Collaboration Server.

Case Name and Storage Path:

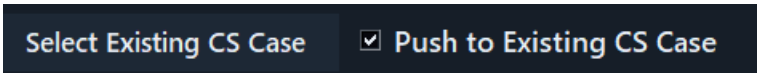


New CS Case

For new cases, with an existing matching Case # not already on the server, enter a Case# or identifier.

Select Existing CS Case

If a case already exists on the collaboration server that you would like to push new data to, check the box for 'Push to Existing CS Case'.

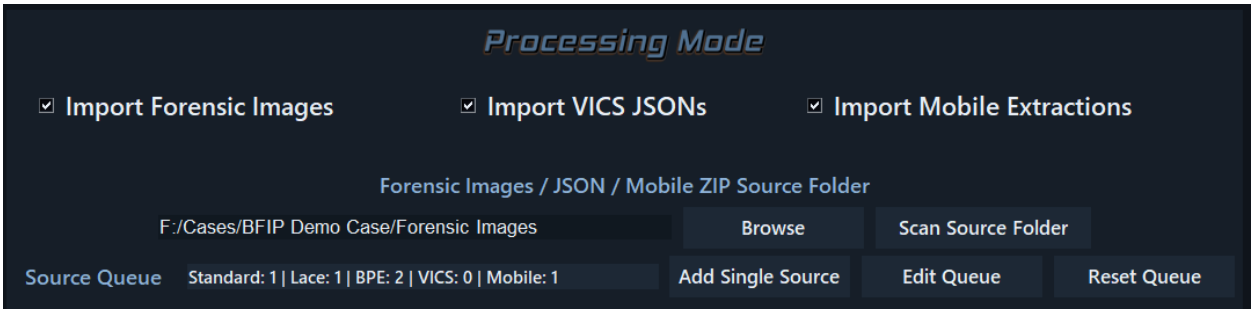


Press the 'Select Existing CS Case' button. BFIP will query the configured collaboration server for a list of existing Case IDs and return a selection window for the user to confirm which existing case they would like to push new data to.

Griffeye CS Storage Path

Under 'Storage Path', designate an available folder or mapped drive where carved data and JSONs generated from your cases forensic images can be temporarily saved to.

Advanced Source Queue Setup



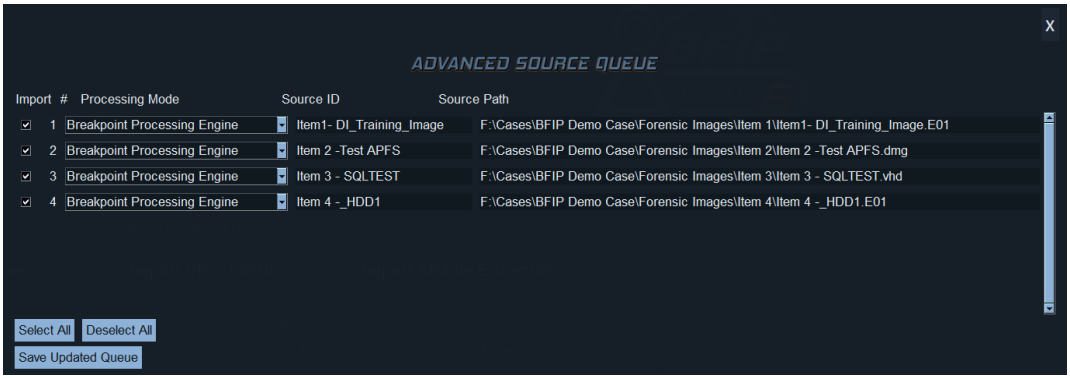
Advanced Source Setup offers granular control over source IDs, processing modes, and source paths. It supports mixing forensic images, VICS JSON packages, and mobile extraction ZIPs in a single queue, including sources from multiple folders or drives.

Instructions:

1. ### Forensic Images/JSON/Mobile ZIP Source Folder:
   Select the parent folder where your forensic images, JSON packages, and/or mobile extraction ZIP archives are stored.

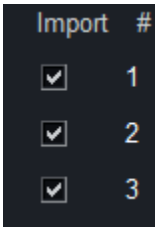
2. ### Scan Source Folder
   BFIP will intelligently search the specified folder and subfolders for supported forensic images, VICS JSON packages, and mobile extraction ZIP archives. Located sources will be displayed in a generated queue configuration window where you may confirm via checkbox which sources to include for processing.

3. Advanced Source Queue Configuration Window

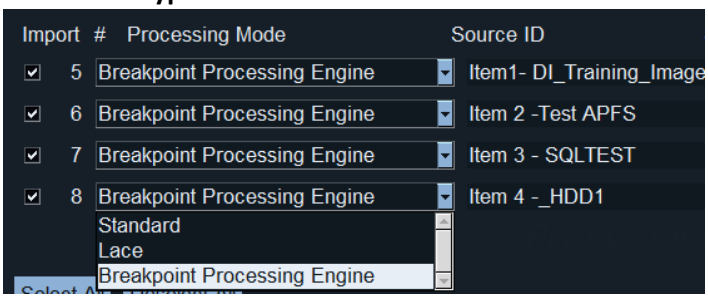


This window will have primary fields available to customize for each located source:

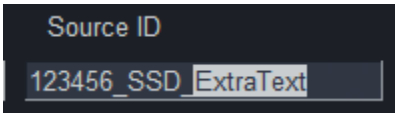
Import Checkbox: Place a checkbox next to any source you want to be included in the carving/import process. Any item that is unchecked will be completely removed from the current queue and no further processes will be conducted on it.



Processing Mode: Select the preferred processing mode/engine for the specific source. A mix of processing modes can be used in the same queue depending on the needs of the examiner and source type.



Source ID: Source IDs are initially auto-generated from the source filename with the extension stripped, but can be individually edited and customized.



Source Path: The unique source path for each file will be shown in this field automatically.
**While this field can be manually adjusted by the user, it is highly recommended you use the automatically generated value.*

4. Add to Queue

After customizing processing mode selections, Source ID selection, etc., select 'Add to Queue', and the Advanced Source Queue Status bar will update to reflect the added sources.

5. Adding additional files to queue (Optional):

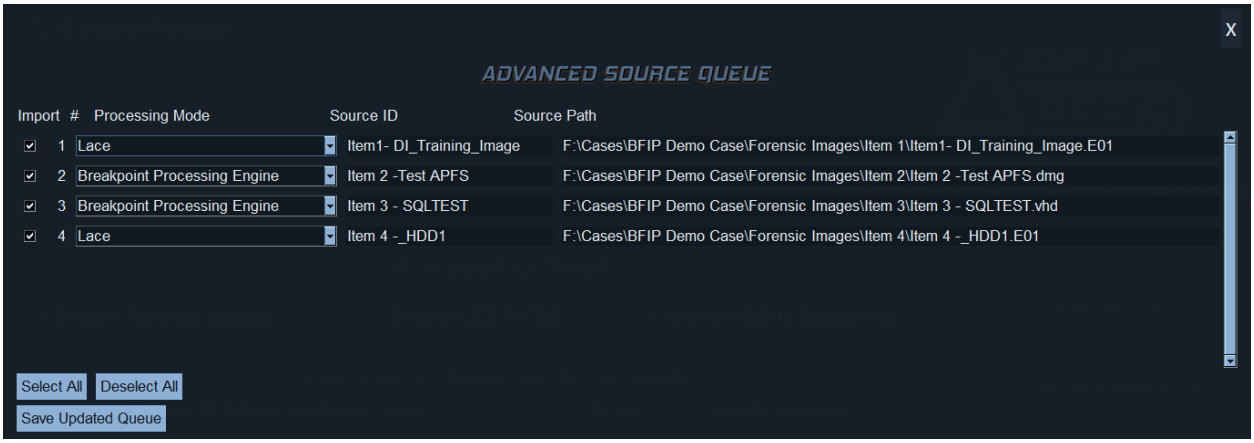
The examiner may repeat this workflow by scanning additional source folders or using Add Single Source. This allows sources scattered across several folder locations to be stacked into a single queue.

6. Add Single Source

Use Add Single Source to select an individual VICS JSON, supported forensic image, or mobile extraction ZIP and add it directly to the advanced source queue. This can be repeated as needed and can be mixed with sources found through bulk source scanning.

7. Edit Queue

This opens a menu containing all sources and current settings already added to the queue. Existing Source IDs, Processing Modes, and paths can be reviewed or changed from here.



8. Reset Queue:

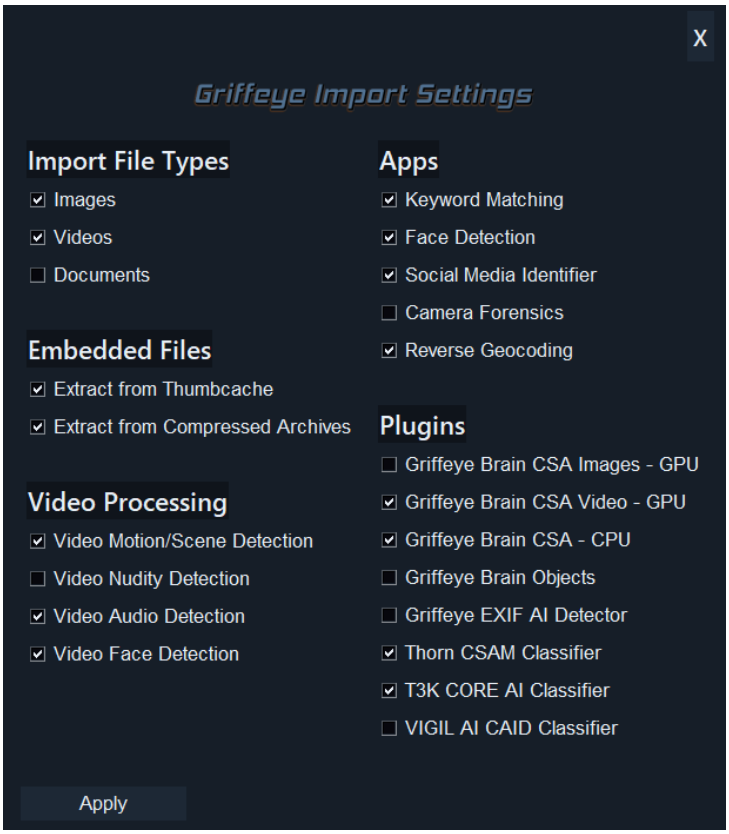
This function completely clears the Advance Source Queue.

Griffeye Import Settings

Griffeye Import Settings Menu

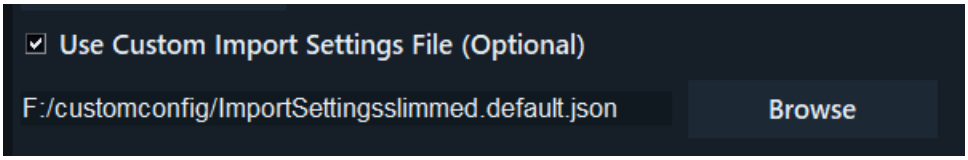
The 'Griffeye Import Settings' Menu allows for direct UI control over several of the most common settings, apps, and plugins that you may want to enable/disable. Associated plugins must be preactivated manually within Griffeye prior to using this interface however. Most apps are now preactivated by default, however Brain plugins will require associated installers and Forensic Market activation.

> **Note in order to ensure these settings are committed to Griffeye you must explicitly open this menu and click 'Apply' with your preferred options set. Otherwise, Griffeye will default to using the configuration settings from your last import job.*



Custom Import Settings JSON

If you prefer to configure specific import settings beyond those available in the BFIP menu options, you can optionally specify a custom Griffeye Import Settings JSON file. This will override any Griffeye Import Settings specified in the menu.



Starting Functions

Start

Run selected processing engines/file carving, conduct Griffeye case creation, and import forensic images, generated VICS packages, and/or queued JSON files into Griffeye as applicable.

**Griffeye DI Pro/Advanced or Griffeye Processing Engine License Required or Griffeye CS(Ops/Enterprise).*

Carve Only

Conducts carving/extraction using Breakpoint Processing Engine and Breakpoint Mobile Processing Engine based on configured queue sources and settings.

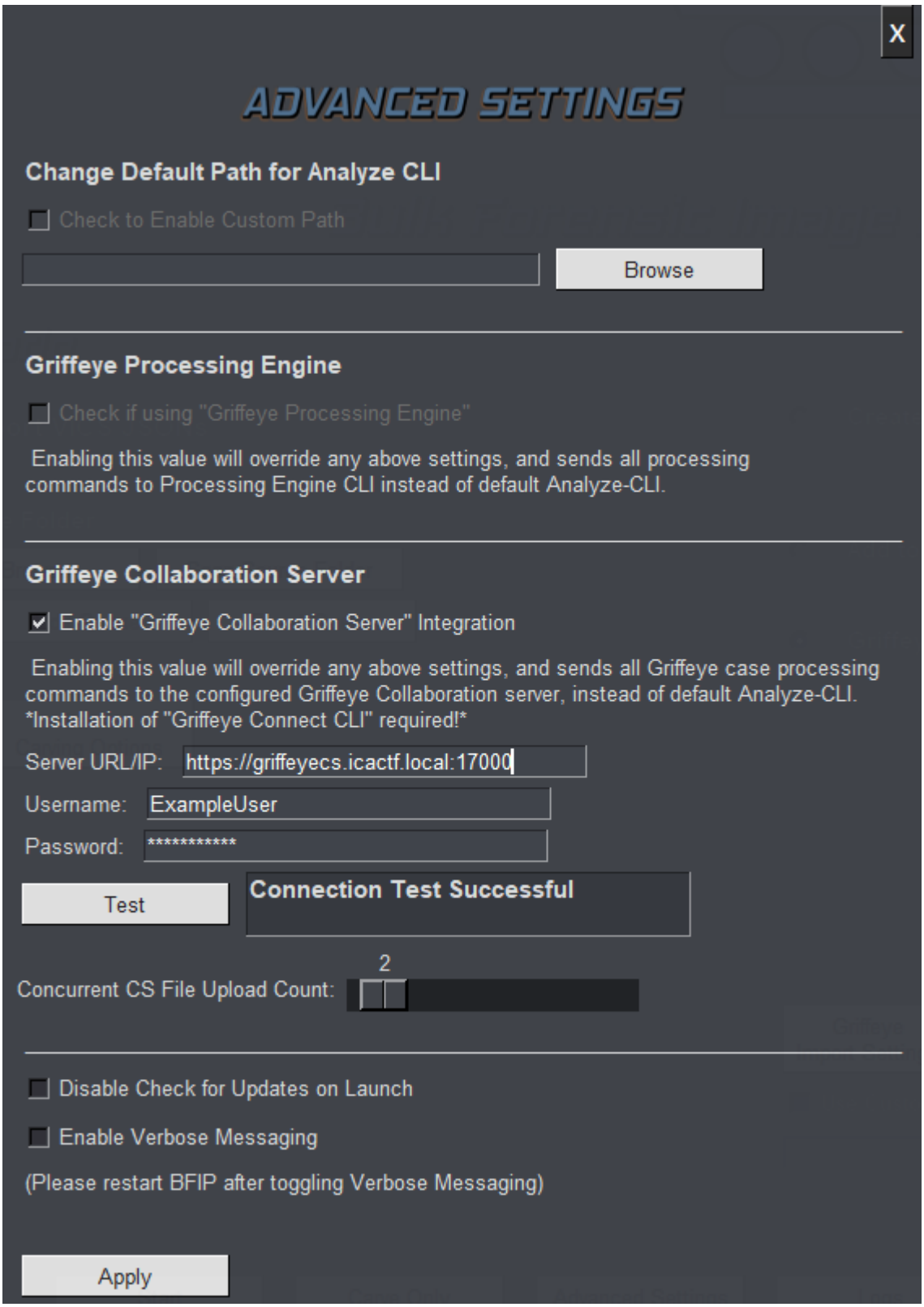
Generates JSON Packages of Carved Content.

Conducts automated extraction from supported forensic image file systems and mobile extraction ZIPs, then generates VICS JSON packages from recovered files and metadata.

Does not automatically pass recovered files/JSON to Griffeye.

No Griffeye License Required.

Advanced Settings



Change Analyze CLI Path:

BFIP looks for the file 'magnet-griffeye-cli.exe' in the default install location of:

C:\Program Files\Magnet Forensics\Magnet Griffeye

If you have changed the default install location, this value can be overridden with a new folder location for 'magnet-griffeye-cli.exe'. Both the 'Check to Enable Custom Path', and new folder location must be specified.

Griffeye Processing Engine:

BFIP Supports interfacing with the CLI-only 'Griffeye Processing Engine'. In order to use the Griffeye Processing Engine you must check the associated box under Griffeye Processing Engine. This will direct all Griffeye commands, settings updates, etc. to the Griffeye Processing Engine instead of magnet-griffeye-cli.

Griffeye Operations/Enterprise Collaboration Server Configuration:

To Enable Griffeye Collaboration Server Integration, located and check the activation box under the 'Advanced Settings' menu.

URL

Enter the full URL including the port # for Griffeye CS. (Default typically 17000)

Example: ***https://griffeyecs.ICACSERVER.local:17000***

User Credentials

Enter a username and the associated password for a user account configured on the collaboration server that will be used for authenticating all case creation commands. Ensure the configured user account is properly configured on the collaboration server with the necessary permissions/claims for creating and editing cases.

Test

After entering the credentials, click 'Test' to validate BFIP is able to establish a connection with the configured server, and that the credentials are excepted.

Concurrent CS File Upload Count

The CS Connect CLI's file pushing function can be configured for multiple parallel file uploads depending on desired performance and network conditions. This value can be increased or decreased as desired by the user.

Apply

If the Connection Test is Successful, click 'Apply' to save the configured settings. The credentials will be encrypted, and securely stored on the local workstation for later recall the next time BFIP is launched.

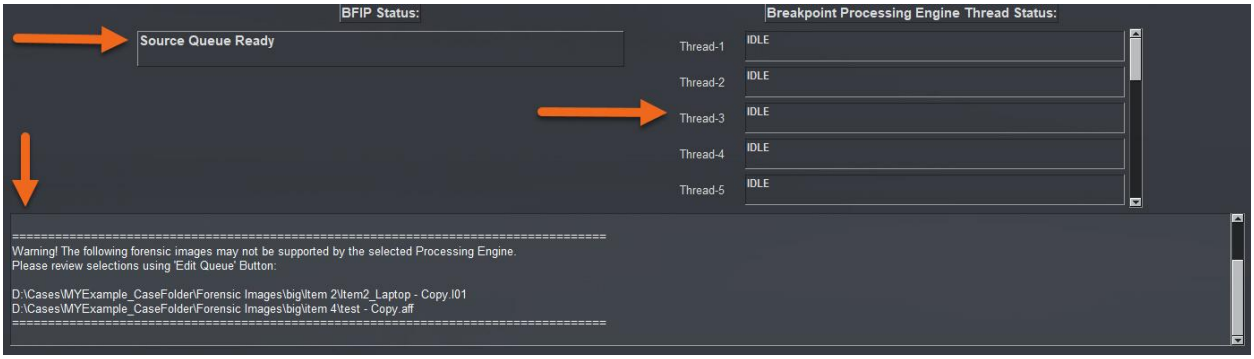
Verbose Messaging:

Enabling verbose messaging increases status messages in console and enables additional debug logging.

Output Window and Status Bar

Griffeye utilizes an integrated output window. Various confirmation messages, processing status, errors, etc. will be printed here for reference.

Directly above the window is a dynamic status bar that will occasionally update with carving status, progress indicators and completion percentages.

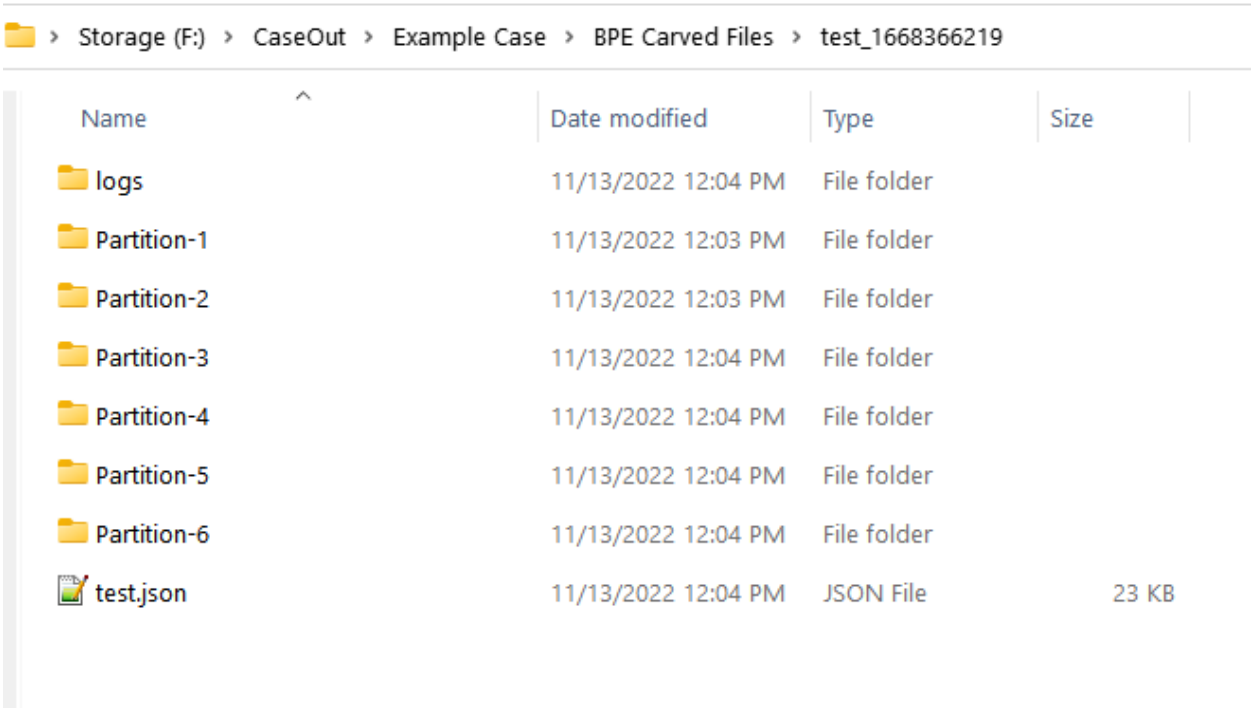


Post Import

Breakpoint Processing Engine

Data Output Locations:

Following a carve or complete carve and import process using Breakpoint Processing Engine or Breakpoint Mobile Processing Engine, recovered data, logs, and generated JSON packages for each Source ID are stored inside the case output folder under BPE Carved Files. Each processed source receives its own subfolder containing exported files, generated VICS JSON, and source-specific logs.

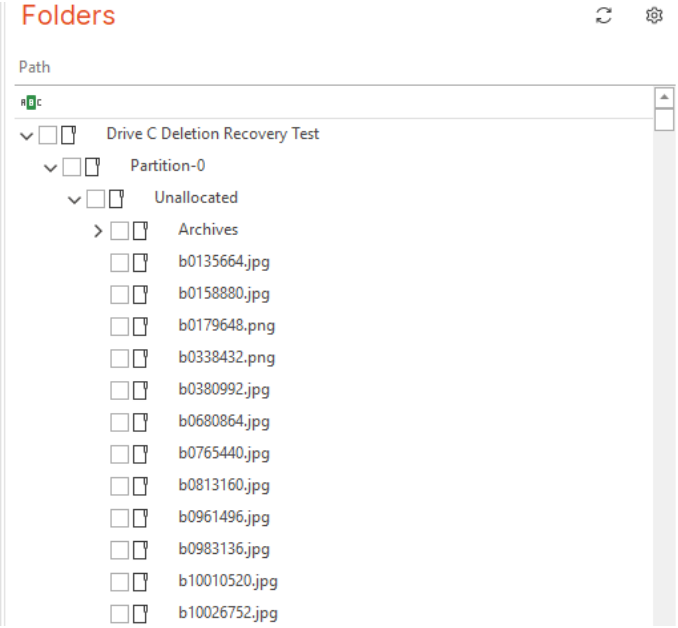


Mobile ZIP Post Griffeye Import

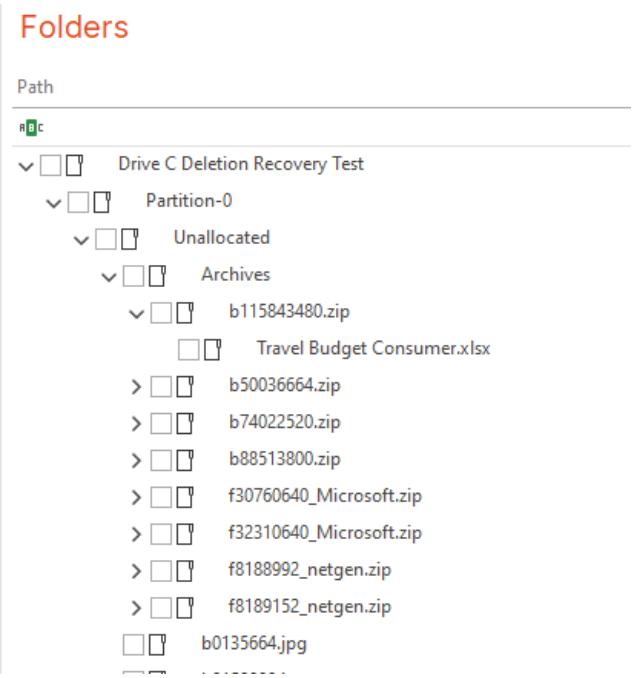
Mobile ZIP outputs are imported as VICS packages. Files recovered from the original ZIP retain their original mobile extraction file paths where available. Files recovered from embedded locations such as databases, cache files, Office documents, or PDFs are written as separate media records with available MD5 and file size metadata.

BPE Post Griffeye Import

If you elected to use the standard '*Start*' option, the data generated with the Breakpoint Processing Engine will automatically be imported into a Griffeye case. For data that was recovered, it will appear under the folder path nomenclature of '**[sourceID]/[Partition-#]/[folderpath]'**



If you elected to both carve and expand archive files, the expanded archives and their respective files will be seeded under an additional subfolder named '**Archives'**, followed by the path to the original parent Archive file.



File MetaData Unallocated

Data recovered and imported using the Breakpoint Processing Engine will fill several metadata fields based on the limited data that can be recovered from unallocated content.

These will include at least:

SourceID: Based on name derived from original forensic image.

Mime Type: Identified File Type Based on File Recovery Process

File Name: Does not reflect original file name prior to deletion. This is named based off the physical location recovered on disk.

Physical Location: The first physical sector where the file was located.

If the file was extracted from a recovered archive, this value will reflect the physical sector of the parent archive file

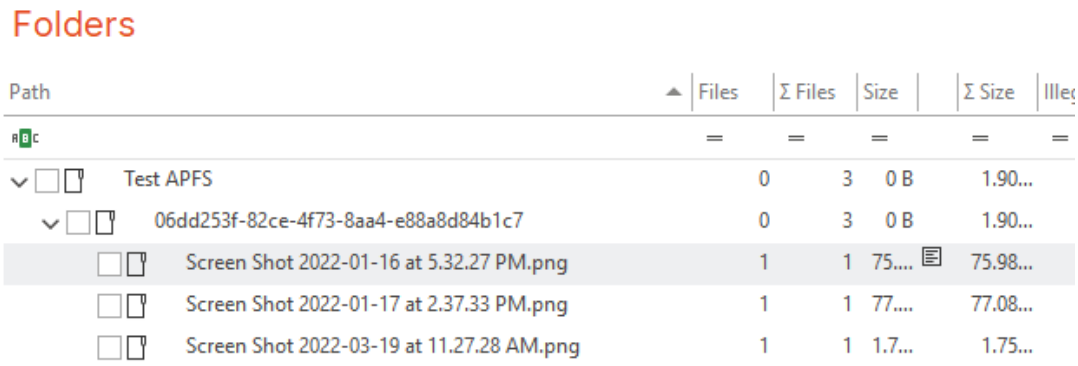
Source ID	Mime Type	File Name	Physical Location	Unallocat...	File Path
Drive C Deletion Recovery Test	image/jpeg	f41871744.jpg	41871744	☑	Drive C Deletion Recovery Test\Partition-0\Unallocated\f41871744.jpg
Drive C Deletion Recovery Test	image/bmp	f41042120.bmp	41042120	☑	Drive C Deletion Recovery Test\Partition-0\Unallocated\f41042120.bmp
Drive C Deletion Recovery Test	image/jpeg	f42084864.jpg	42084864	☑	Drive C Deletion Recovery Test\Partition-0\Unallocated\f42084864.jpg
Drive C Deletion Recovery Test	image/webp	f40521024.avi	40521024	☑	Drive C Deletion Recovery Test\Partition-0\Unallocated\f40521024.avi

BPE Post Griffeye Import – Live Files

If you process live/allocated files using Breakpoint Processing Engine, any active/allocated files located in the file-system will be extracted and imported into Griffeye with their original associated metadata. They will appear under the folder path nomenclature of '**[SOURCEID]/[Partition#]/FOLDERS]**

For APFS Data it will include APFS Container information as part of the path:

'**[SOURCEID]/[APFS Container-GUID]/[CONTAINER-FILES/FOLDERS]**



File MetaData Live Files

Data recovered and imported using the Breakpoint Processing Engine will recover and fille common metadata fields as located in the file-system.

These will include at least:

SourceID: Based on name derived from original forensic image.

Mime Type: Identified File Type Based on File Recovery Process

File Name: Original File Name as located in APFS file-system.

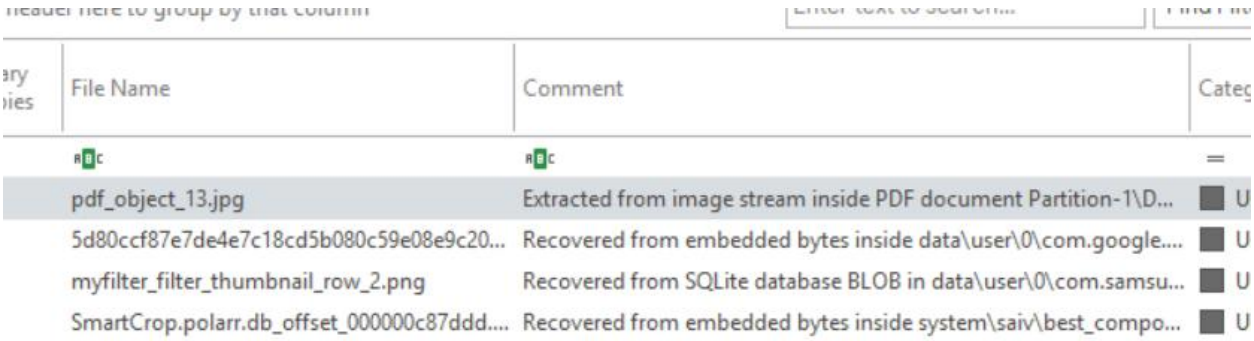
File Path: Original File Path as located in APFS file-system.

MAC Timestamps: Original modified, created, and accessed timestamps as located in file-system.

Source ID	File Name	File Path	Created Date	Last Write Time	Last Accessed
Test APFS	Screen Shot 2022-01-16 at...	Test APFS\06dd253f-82ce-4f73...	1/16/2022 5:32:33 PM	1/16/2022 5:32:33 PM	10/28/2022 8:20:43 AM
Test APFS	Screen Shot 2022-01-17 at...	Test APFS\06dd253f-82ce-4f73...	1/17/2022 2:37:39 PM	1/17/2022 2:37:39 PM	10/28/2022 8:20:43 AM
Test APFS	Screen Shot 2022-03-19 at...	Test APFS\06dd253f-82ce-4f73...	3/19/2022 11:29:56 AM	3/19/2022 11:29:56 AM	10/28/2022 8:20:43 AM

Additional Optional Metadata Fields:

Comments: If BFIP located and extracted media embedded within databases and other similar structures, additional notations may be located here. See reference for more details.



File Name	Comment	Categ
pdf_object_13.jpg	Extracted from image stream inside PDF document Partition-1\D...	U
5d80ccf87e7de4e7c18cd5b080c59e08e9c20...	Recovered from embedded bytes inside data\user\0\com.google....	U
myfilter_filter_thumbnail_row_2.png	Recovered from SQLite database BLOB in data\user\0\com.samsu...	U
SmartCrop.polarr.db_offset_000000c87ddd....	Recovered from embedded bytes inside system\saiv\best_compo...	U

BFIP Embedded Recovery Comments Reference

Purpose: This reference explains the Comments values BFIP may write into VICS JSON for files recovered from inside another file, database, document, or storage structure.

Note: Standalone files recovered directly from a mobile ZIP or forensic image normally do not receive these embedded recovery comments.

Comment Text	Plain-Language Meaning
Recovered from SQLite database BLOB in [path].	Media was stored directly inside a SQLite database field as binary data.
Recovered from base64-encoded SQLite text in [path].	Media was stored inside a SQLite text field as base64 text, then decoded by BFIP.
Recovered from compressed SQLite value in [path].	Media was stored inside a SQLite field in compressed form and was decompressed before export.
Recovered from embedded bytes inside SQLite database [path].	BFIP found media signatures inside a larger SQLite value rather than the whole database value being only one clean media file.
Recovered from SQLite database content in [path].	Generic SQLite recovery note when BFIP knows the item came from SQLite but the exact storage pattern is broader or less specific.
Recovered from embedded bytes inside LevelDB/RocksDB file [path].	BFIP found recoverable media bytes inside a LevelDB or RocksDB storage file.
Extracted from image stream inside PDF document [path].	BFIP extracted an image object embedded inside a PDF file.
Extracted from embedded media inside Office document [path].	BFIP extracted embedded media from an Office document, such as DOCX, PPTX, or XLSX-style containers.
Recovered from embedded content inside document [path].	Generic document-embedded recovery note when the document source is known but does not match the more specific PDF or Office wording.
Recovered from embedded bytes inside [path].	BFIP found media signatures inside another file or container, but the source type was not one of the more specific database/document categories.
Recovered from embedded mobile extraction content.	Fallback note for mobile embedded recovery when BFIP recovered embedded content but does not have a specific container path available.

Breakpoint Processing Engine API-Mode

BFIP's Breakpoint Processing Engine and Breakpoint Mobile Processing Engine can be controlled and executed via CLI and a structured JSON API. This allows integration with other tools or custom scripts/automations. The API functions by calling the BFIP executable followed by supported arguments and a path to a supported JSON file.

Example: **BFIP.exe -JF "d:\myBPESources.json"**

Further details and the most up to date example of the recommended JSON API spec is available here:

https://github.com/breakpointforensics/BPE_API_DATAMODEL

Logs and Troubleshooting

Logging for operations initiated with BFIP are maintained in 3 primary locations.

BFIP Logs

BFIP specific logs, such as Breakpoint Processing Engine, import status, etc. are logged in a BFIP's AppData folder along with saved user preferences. These can be directly accessed by clicking on the 'Logs' button at the bottom of the main BFIP interface. A new log is generated for each calendar day.

Default:

C:\Users\[USERNAME]\AppData\Local\BreakpointForensics\BFIP\Logs

Breakpoint Processing Engine - Source Specific Logs

Additional BFIP source-specific logs generated by Breakpoint Processing Engine and Breakpoint Mobile Processing Engine are logged in the case output folder under:

[CaseFolder]\BPE Carved Files\[SOURCEID]\Logs

Griffeye Logs

Once BFIP passes the import parameters to the Griffeye CLI during the Griffeye Processing phase, all Griffeye related messages, errors, and status will be reflected in the normal Griffeye log folder/file.

Default:

C:\ProgramData\Magnet Forensics\Magnet Griffeye\Logs