ADF Data Container Structure

Introduction

When acquiring data from a target device, the ADF tools create a data container. This document describes the structure of that data container.

What is the Data Container?

Data containers are created to store the files and metadata found on target devices (a.k.a. data sources). Their goals are to make it easy to collect a variety of data, store it in a way that's easy to transport, and be searched and processed.

Structure

The data acquired from a data source is saved in a folder with the following files and folders:

Acquisition metadata

- metadata.zip
    - This file contains information about the acquisition and the data source
        - info
            - data_container.json
            - acquisition_information.json
            - data_source.json
        - log
            - acquisition logs

Files from a remote agent data source

- files.zip: contains the files acquired from the data source. The following file properties are preserved inside this archive: paths, file names and timestamps.
- embedded.zip: contains the files extracted or carved out of containers.
- inaccessible.zip: contains the file names of files that were inaccessible during an acquisition (usually, it's because they were blocked by the Operating System).
- files_meta.zip: contains json files called file_meta_NNNN.json which save the filesystem properties of the files saved in files.zip. Up to 1000 records are saved in each json file.

Files from an iOS device acquisition

- afc_backup.zip: contains the files obtained from the Apple File Conduit protocol.
- iTunes_backup (folder): contains the files obtained from an iTunes backup.
- iTunes_encrypted_backup (folder): contains the files obtained from an encrypted iTunes backup.

Files from an Android device acquisition

- adb_backup.zip: contains the files obtained from an adb backup.
- shared_backup.zip: contains the files obtained from the Android shared storage space.

Files from a screen capture

- sc_files (folder): contains the screenshots and video recording from the video stream or mirrored device.
    - screen_capture.json: contains the metadata for all the collected information.

Digital artifacts

- artifacts_raw.zip: contains information obtained from API/system calls executed directly on the target device.
- artifacts.zip: contains json files with the digital artifacts extracted from the data source.

Load Files

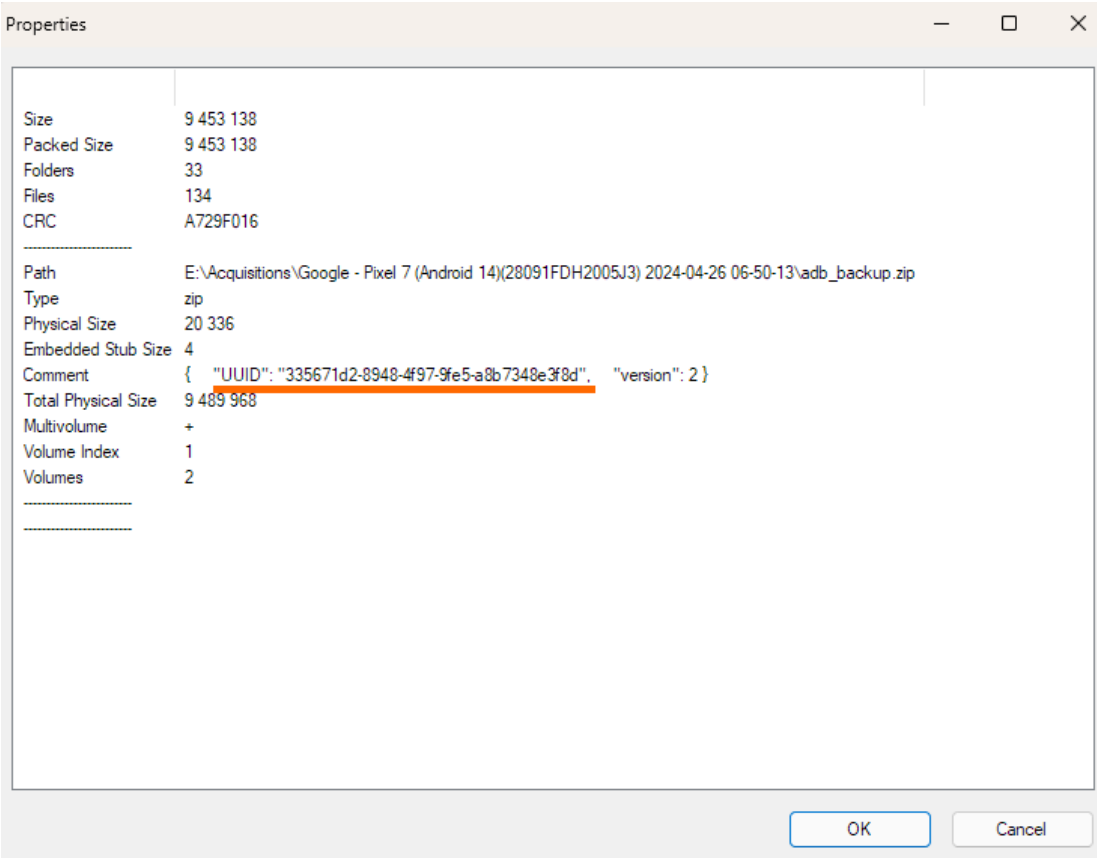
- load_files (folder): contains the dat load files.

The zip files are all multi-volume zip files (except metadata.zip) with a maximum file size of 4GB.

Forensic Integrity

The data container includes some built-in features that help with the forensic integrity of the data.

- All the zip files that contain the acquired data are hashed and the hash values are saved in the acquisition_information.json file (see below). This makes it possible to verify the integrity of these files over time.
- For folders, the number
- The unique identifier of the data container (defined in metadata.zip/info/data_container.json/id) is also saved in each zip file's metadata in the Comment section as the UUID field. This makes it possible to verify which files belong to the data container.



Zip file metadata showing the UUID of the data container

metadata.zip

Several log files are created during the imaging process and saved in the metadata.zip file. Here are examples of such files.

data_container.json

This file contains information about the data container where the acquisition data is saved.

```
{
    "$id": "/schema/cm/v1/data_container",
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "id": "335671d2-8948-4f97-9fe5-a8b7348e3f8d",
    "name": "Google - Pixel 7 (Android 14)(28091FDH2005J3) 2024-04-26 06-50-13"
    "creationDate": "2024-04-26T10:50:14.011682Z",
    "createdByAppName": "ADF Digital Evidence Investigator",
    "type": "Acquisition"
}
```

acquisition_information.json

This file contains information about the acquisition process.


```
{
  "caseInformation": {
    "name": "Case AAE-43321",
    "location": "123 main street, Washington DC 20001"
  },
  "acquisitionDetails": {
    "id": "1234567890",
    "createdByAppName": "ADF ADF Digital Evidence Investigator",
    "createdByAppVersion": "6.1.0",
    "startDate": "2025-02-17T12:00:00.0000000Z",
    "finishedDate": "2025-02-17T13:00:00.0000000Z",
    "status": "Completed",
    "acquisitions": [
        {
            "method": "Android Shared Data",
            "status": "Interrupted",
            "formatDetails": {"format": "zip", "type": "logical"},
            "files": [
                {"filename": "shared_backup.zip", "sha256":
"5e6e49171c0a25f36c60ce64b567f2150e359de9c1f26da245adef9d380b6217"}
            ]
        },
        {
            "method": "Android Debug Bridge",
            "status": "Incomplete",
            "formatDetails": {"format": "zip", "type": "logical"},
            "files": [
                {"filename": "adb_backup.zip", "sha256":
"4e6e49171c0a25f36c60ce64b567f2150e359de9c1f26da245adef9d380b6217"}
            ]
        },
        {
            "method": "Android Data Dump App",
            "status": "Completed",
            "formatDetails": {"format": "zip", "type": "logical"},
            "files": [
                {"filename": "artifacts_raw.zip", "sha256":
"93eb0f41c8578a524554c9fa2ff84a11920785ea073fe5d5a1c852d807455e55"}
            ]
        },
        {
            "method": "Android Debug Bridge Casting",
            "status": "Completed",
            "formatDetails": {"format": "flat file", "type": "logical"},
            "folders": [
                {"folderName": "sc_files", "numberOfFiles": "7", "totalSize": 445321}
            ]
        },
        {
            "method": "Load Files",
            "status": "Completed",
```

```
            "formatDetails": {"format": "flat file", "type": "logical"},
            "folders": [
                {"folderName": "load_files", "numberOfFiles": "4", "totalSize": 7832}
            ]
        }
    ]
  }
}
```


data_source.json

This file contains information about the target device.


```
[
    {
        "$id": "/schema/cm/v1/data_source",
        "$schema": "https://json-schema.org/draft/2020-12/schema",
        "dataContainerId": "2a52f03e-ab95-481e-b437-0215736eeff5",
        "drive": {
            "location": "Internal",
            "protocol": "Apple Fabric",
            "sectorSize": 4096,
            "size": 251000193024
        },
        "mountName": "/dev/disk0",
        "name": "APPLE SSD AP0256Q",
        "timeZone": "US/Pacific"
    }
]
```


Load Files

Load Files can be created during an acquisition if the option is selected.

A load file is created for each one of these record types:

- Application
    - Installed Application
- Communication
    - Call
    - Email
    - Message
    - Saved Contact
    - Voicemail
- User Data
    - Calendar
    - Note

Load File Structure

The created load files use the following specifications:

- .dat file extension
- UTF8 encoding
- Header row that defines the exact properties for each record
- Column definition
    - DOCID: mandatory column containing the unique identifier of the record
    - ParentDocID: used to connect a child record to its parent record (for example an attachment to the message it comes from)
    - DocType: the record type
    - DocumentFolderPath: references an external file (such as a message content, or an attachment). Note that this field contains a path that is relative to the location of the load file
- Reserved characters
    - Column delimiter: ASCII 20
    - Quote: ASCII 254
- Date format: all date/time are in UTC and use this format YYYY-MM-DDTHH:MM:SS+00:00
- Sub-folders
    - RECORD_TYPE_NAME_txt_content_N: containing the textual content of the record. For example: message_txt_content, etc.
    - RECORD_TYPE_NAME_bin_content_N containing the binary content of the files associated with the records. For example: message_bin_content, etc.
    - Each sub-folder will only contain up to 10000 files