"Database audit files ETL" User manual

Table of Contents

2.1 Installation 5

2.2 System Requirements 5

4.1 config.properties 7

4.2 Extract\_and\_load.py 10

4.2.1 Usage: 10

4.2.2 Example, Script Call from Python: 10

4.2.3 Example 1, Script Call from console: 10

4.2.4 Example 2, Script Call from console: Day and month will be calculated by module according [EXTRACTION] and [COMPRESS] configuration section defined in config.properties 10

4.2.5 Dependencies: 11

4.3 Activity\_report\_generator.py 11

4.3.1 Usage Examples: 11

4.3.2 Dependencies: 11

4.4 Extractor.py 12

4.4.1 Usage Examples: 13

4.4.2 Dependencies 13

4.5 Loader\_by\_db.py 13

4.5.1 Usage Examples: 14

4.5.2 Dependencies 14

4.6 Compress\_aud\_files.py 15

4.6.1 Usage Examples: 15

4.6.2 Dependencies: 15

4.7 Compress\_db\_aud\_files.py 15

4.7.1 Usage Examples: 15

4.7.2 Dependencies: 16

6.1 Glossary 18

6.2 Validation modules 18

6.3 Content example of .aud file 19

6.4 Content example of Activity report 20

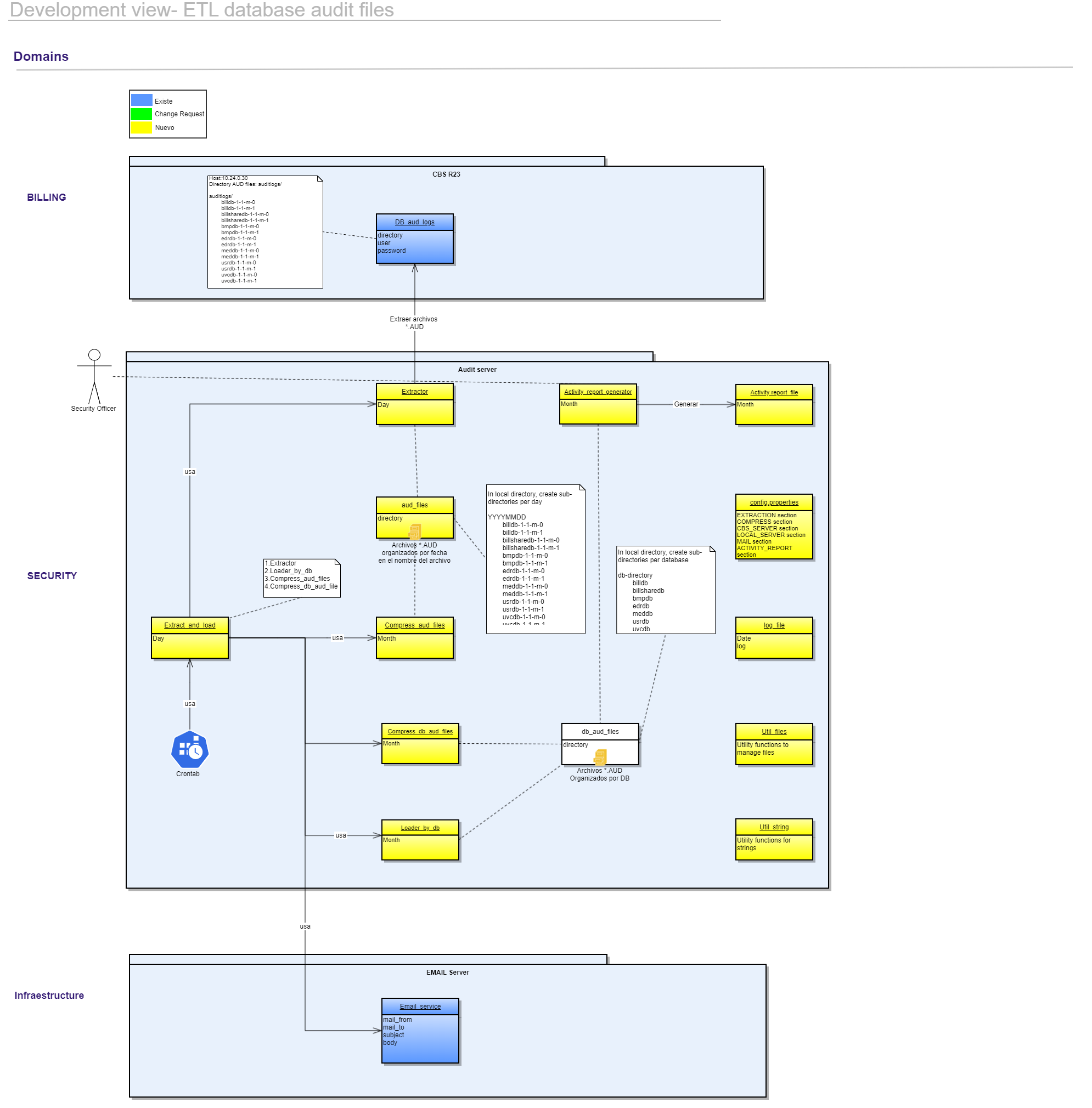
6.5 Content example of Extract\_and\_load module 22

6.6 References 23

# 1. Introduction

Welcome to the user manual for "Database audit files ETL". This manual is designed to help you get started with using the application effectively. Whether you're a new user or looking to explore advanced features, this guide will provide you with the necessary information.

A useful overview from "Database audit files ETL" describes the next development view diagram



# 2. Getting Started

## Installation

Please, follow installation guide published in this link [992-"Database audit files ETL" Installation manual](https://confluence.tigo.net.bo/pages/viewpage.action?pageId=154510753)

## System Requirements

* Operating System: Windows 10 or higher. Linux that supports Python 3.12.3
* Processor: Intel Core i5, CPU 1.60GHz - 1.80GHz
* RAM: 16.0 GB
* Disk Space: 5 GB at least.
* **Python 3.12.3 installed.**
  + You could verify Python version get in python console for example
    - *>py*  
      *Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32*  
      *Type "help", "copyright", "credits" or "license" for more information.*
* Internet Connection: Just for Python installation modules
* Installed Python modules must be (read Apendix - "Validation modules" subtitle for more details):
  + *1. os*  
    *2. logging*  
    *3. configparser*  
    *4. re*  
    *5. datetime*  
    *6. traceback*  
    *7. glob*  
    *8. sys*  
    *9. shutil*  
    *10. hashlib*  
    *11. smtplib*  
    *12. paramiko*  
    *13. tarfile*  
    *14. base64*  
    *15. from Util\_string import getAuditedActivity, getDataBetween, getObjectNameFromQuery*  
    *16. from scp import SCPClient*  
    *17. from Util\_files import deleteDirContent, createSSHClient, read\_config, createAudFilesInLocalDir, writeScriptsChecksumInLog, getFilesQuantityInRemoteDir, getParamikoSSHCLient, getFilesQuantityInDir, compressFiles*  
    *18. from Extractor import extractor*  
    *19. from Loader\_by\_db import loader\_by\_db*  
    *20. from Compress\_aud\_files import compress\_aud\_files*  
    *21. from os.path import basename*  
    *22. from email.mime.application import MIMEApplication*  
    *23. from email.mime.multipart import MIMEMultipart*  
    *24. from email.mime.text import MIMEText*  
    *25. from email.utils import COMMASPACE, formatdate*

# 3. User Interface Overview

Currently Python ETL scripts don`t has user interface.

The only way to interact with Python ETL scripts is by console.

Each script has a main feature that will be described in the next section.

# 4. Features

Each Python script will be described has follow:

1. **config.properties**: Contains system configuration, written in a format that the ***configparser*** module can parse.
2. **Extract\_and\_load.py**: Main features scripts are extract \*.aud files from remote server, organize \*.aud files by database and compress old \*.aud files by month. Uses Extractor.py, Loader\_by\_db\_py, Compress\_aud\_files.py, Compress\_db\_aud\_files.py scripts.
3. **Activity\_report\_generator.py**: Main feature is generating a montly report file. Parse each \*.aud file in directory organized by database.
4. **Extractor.py**: For copy \*.aud files from remote CBS R23 server using SSH connection (through python paramiko module) and applying SCP Linux command by python SCPCliente module.
5. **Loader\_by\_db.py**: Copy \*.aud files from *local\_dir\_audit\_files* to *local\_dir\_audit\_files\_by\_db* and organize them by database.
6. **Compress\_aud\_files.py**: Compress all \*.aud files in *local\_dir\_audit\_files* directory that belong a defined month. This month could be a parameter otherwise will be calculated according to [COMPRESS] configuration section.
7. **Compress\_db\_aud\_files.py**: Compress all \*.aud files in *local\_dir\_audit\_files\_by\_db* directory that belong a defined month. This month could be a parameter otherwise will be calculated according to [COMPRESS] configuration section.
8. **Util\_files.py**: Contains functions for file and directory management. These functions are used by all the scripts previously explained in this list, except config.properties.
9. **Util\_string.py**: Contains functions for string management. These functions are used by *Activity\_report\_generator.py* script.
10. **Validate\_modules.py**: Useful for installation stage. Read Apendix - "Validation modules" subtitle for more details.

## config.properties

This files must be in *local\_dir\_scripts* directory, parameter defined in the same config.properties file.

Contains system configuration, written in a format that the ***configparser*** module can parse. It has de following sections:

1. [EXTRACTION]: Configurations related to extraction process. When to extract next aud files, if it is necessary delete destiny directory and where put calculated checksum files are defined in this section.
2. [COMPRESS]: Parameters related to compression feature like wich month \*.aud files must be compressed in *local\_dir\_audit\_files* and *local\_dir\_audit\_files\_by\_db.*
3. [CBS\_SERVER]: Configurations parameters related to remote CBS server is defined. Host, por. user, password, \*.aud files directory and sub-directories for each node are defined in this section.
4. [LOCAL\_SERVER]: Local directory for \*.aud where copied from remote server, local directory for organize \*.aud files for database, local directory for python scripts files and local directory for activity reports are defined in this section.
5. [MAIL]: Configuration related to send a e-mail.
6. [ACTIVITY\_REPORT]: Configuration related to report generation.

Almost each parameter has comments to describe its functionality. The initial content of this file is:

|  |
| --- |
| [EXTRACTION]  # AUD files will be extracted N days after they are created in remote server  old\_files\_in\_days\_to\_be\_extracted **=** 2  # 1: Delete destiny dir content before copy new data. 0: Does not delete destiny dir content before copy new data  delete\_destiny\_dir\_content **=** 1  # 1: Generate chesksum files. 0: Does not generate checksum files  generate\_checksum\_files **=** 0  # 1: Generate chesksum logs. 0: Does not generate checksum logs  generate\_chesksum\_log **=** 1    [COMPRESS]  # Compress N months age imported AUD files  compress\_imported\_aud\_files\_n\_months\_after **=** 2  # Compress N months age organized by DB AUD files  compress\_aud\_files\_organized\_by\_db\_n\_months\_after **=** 2  # Compress log files where created last N month  compress\_logs\_files\_n\_months\_after **=** 2  # 1: Compress activity reports inmediatly. 0: Does not compress  compress\_activity\_report\_inmedtaly **=** 2    [CBS\_SERVER]  host **=** 12.34.5.06  port **=** 22  user **=** auditfiles  password **=** XXXYYZZZ  cbs\_base\_dir\_audit\_files **=** **/**auditlogs  cbs\_sub\_dir\_list\_audit\_files **=**  billdb**-**1**-**1**-**m**-**0  billdb**-**1**-**1**-**m**-**1  billsharedb**-**1**-**1**-**m**-**0  billsharedb**-**1**-**1**-**m**-**1  bmpdb**-**1**-**1**-**m**-**0  bmpdb**-**1**-**1**-**m**-**1  edrdb**-**1**-**1**-**m**-**0  edrdb**-**1**-**1**-**m**-**1  meddb**-**1**-**1**-**m**-**0  meddb**-**1**-**1**-**m**-**1  usrdb**-**1**-**1**-**m**-**0  usrdb**-**1**-**1**-**m**-**1  uvcdb**-**1**-**1**-**m**-**0  uvcdb**-**1**-**1**-**m**-**1      # Local directories  [LOCAL\_SERVER]  # Directory where \*.aud files will be copied in extraction process  local\_dir\_audit\_files **=** **/**home**/**arcsight**/**auditCBS  # Directory where \*.aud files will be copied from local directory local\_dir\_audit\_files an organized by database  local\_dir\_audit\_files\_by\_db **=** **/**home**/**arcsight**/**auditCBS\_Processed  # Directory Python scripts files  local\_dir\_scripts **=** **/**root**/**Scripts  # Directory log files  local\_dir\_logs **=** **/**root**/**Scripts**/**logs  # Directory report files  local\_dir\_reports **=** **/**root**/**Scripts**/**reports    [MAIL]  mailFrom **=** cbsaudit@domain.com  #mailTo = informationsecurity@domain.com  mailTo **=** gonzalesv@domain.com  mailSubject **=** "CBS Audit Trail LOG"  mailBody **=** "Job Success"  mailServer **=** millicomhq.relay.tmes.trendmicro.com  mailPort **=** 25    [ACTIVITY\_REPORT]  # 1: Add summary report at the end of report. 0: Does not add summary report at the end of report  add\_summary\_report **=** 0 |

Code Block 1 config.properties

## Extract\_and\_load.py

This module provides functionalities to extract, load, compress, and send audit database files.

### Usage:

    - To run the main function with optional arguments:  
        $ python.exe Extract\_and\_load.py "extract\_date" "compress\_aud\_month" "compress\_db\_aud\_month"

    - Extract\_and\_load.py script accepts three optional arguments:  
        1. "extract\_date": Date in the format "YYYYMMDD" for data extraction.  
        2. "compress\_aud\_month": Month in the format "YYYYMM" for compressing audit files.  
        3. "compress\_db\_aud\_month": Month in the format "YYYYMM" for compressing database audit files.

    - As result a string 'OK' or 'ERROR' is written into console  
      
    - More processing details are in **extract\_and\_load-{logFileDateStr}.log** file into logs directory (defined in *local\_dir\_logs* parameter in config.properties) .   
      logFileDateStr = datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")  
      For example: extract\_and\_load-20240510145357630753.log

### Example, Script Call from Python:

    >>> from Extract\_and\_load import extract\_and\_load  
    >>> **extract\_and\_load("20240509", "202403", "202404")**  
    OK

### Example 1, Script Call from console:

    $ python.exe Extract\_and\_load.py "20240509" "202403" "202404"

    OK

### Example 2, Script Call from console: Day and month will be calculated by module according [EXTRACTION] and [COMPRESS] configuration section defined in config.properties

    $ **python.exe Extract\_and\_load.py**

    OK

### Dependencies:

    - Extractor module: Provides functionality for data extraction.  
    - Loader\_by\_db module: Provides functionality for loading data into *local\_dir\_audit\_files\_by\_db* directory.  
    - Compress\_aud\_files module: Provides functionality for compressing audit files to save tar files into *local\_dir\_audit\_files* directory.  
    - Compress\_db\_aud\_files module: Provides functionality for compressing database audit files to save tar files into *local\_dir\_audit\_files\_by\_db* directory.  
    - Util\_files module: Provides utility functions for reading configuration and writing checksum in logs.

## Activity\_report\_generator.py

Activity report generator module contains functions to check database audited activities described in database audit files (AUD).

The database audited activity are: Insert, Update, Delete, Truncate and Drop.

The main function is checkForAuditedActivities. This function parse all AUD files in a directory (parameter local\_dir\_audit\_files\_by\_db defined in config.properties file) to search for audited activities.

All data from audited activities founded is saved in a report file named as **activity\_report-{logFileDateStr}.txt.**

logFileDateStr = datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")

A content example of audit file (.aud) is in Apendix - "Content example of .aud file" section.

A content example of report file with all data from each audited activity is in Apendix - "Content example of Activity report".

### Usage Examples:

    1. Generating an activity report for a specific month:  
    >>> from database\_audit\_etl import activity\_report\_generator  
    >>> **activity\_report\_generator("202405")**  
    'OK'

### Dependencies:

    - os  
    - logging  
    - configparser  
    - re  
    - datetime  
    - Util\_string.getAuditedActivity  
    - Util\_string.getDataBetween  
    - Util\_string.getObjectNameFromQuery  
    - Util\_files.writeScriptsChecksumInLog  
    - Util\_files.read\_config  
    - traceback  
    - glob  
    - sys

## Extractor.py

A module for extracting audit files (\*.aud) from external server, copying them locally, and generating checksums.

In remote server in a directory defined (see config.properties) there is a main directory that contains sub-directories like this:  
    auditlogs  
        billdb-1-1-m-0  
        billdb-1-1-m-1  
        billsharedb-1-1-m-0  
        billsharedb-1-1-m-1  
        bmpdb-1-1-m-0  
        bmpdb-1-1-m-1  
        edrdb-1-1-m-0  
        edrdb-1-1-m-1  
        meddb-1-1-m-0  
        meddb-1-1-m-1  
        usrdb-1-1-m-0  
        usrdb-1-1-m-1  
        uvcdb-1-1-m-0  
        uvcdb-1-1-m-1

In each directory there are audit files named as like next example:  
        zengine\_20240503134936581.aud  
        zengine\_20240504012422459.aud  
        ..

The result of extraction will be a new directory (named using format YYYYMMDD) created into a base directory (parameter local\_dir\_audit\_files defined in config.properties file).   
This new directory will have all files copied from one day, for example:  
        
    auditCBS  
        **20240503**  
            billdb-1-1-m-0  
            billdb-1-1-m-1  
            billsharedb-1-1-m-0  
            billsharedb-1-1-m-1  
            bmpdb-1-1-m-0  
            bmpdb-1-1-m-1  
            edrdb-1-1-m-0  
            edrdb-1-1-m-1  
            meddb-1-1-m-0  
            meddb-1-1-m-1  
            usrdb-1-1-m-0  
            usrdb-1-1-m-1  
            uvcdb-1-1-m-0  
            uvcdb-1-1-m-1         
        **20240504**  
            billdb-1-1-m-0  
            billdb-1-1-m-1  
            billsharedb-1-1-m-0  
            billsharedb-1-1-m-1  
            bmpdb-1-1-m-0  
            bmpdb-1-1-m-1  
            edrdb-1-1-m-0  
            edrdb-1-1-m-1  
            meddb-1-1-m-0  
            meddb-1-1-m-1  
            usrdb-1-1-m-0  
            usrdb-1-1-m-1  
            uvcdb-1-1-m-0  
            uvcdb-1-1-m-1        
        ..

Relevant configurations for this module are in [CBS\_SERVER] and [LOCAL\_SERVER] section into config.properties file.

When this module is called directly, details of this task are logged into **extractor-{logFileDateStr}.log** file  
With logFileDateStr = datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")

### Usage Examples:

    1. Copy audit files from an external server to a local directory:  
    >>> from aud\_file\_extractor import copyAudFilesFromExternalServer  
    >>> copyAudFilesFromExternalServer("/source/dir", "/destination/dir", "202405", "subdir1\nsubdir2", "server", "port", "user", "password")  
    'OK'

    2. Calculate checksums for audit files in a directory:  
    >>> from aud\_file\_extractor import checksumFiles  
    >>> checksumFiles("/destination/dir", "202405", generate\_checksum\_files='1', generate\_chesksum\_log='1')  
    'OK'

    3. Extract audit files from an external server and generate checksums:  
    >>> from aud\_file\_extractor import extractor  
    >>> **extractor("202405")**  
    'OK'

### Dependencies

    - shutil  
    - os  
    - logging  
    - datetime  
    - glob  
    - re  
    - SCPClient (from scp)  
    - traceback  
    - hashlib  
    - Util\_files.deleteDirContent  
    - Util\_files.createSSHClient  
    - Util\_files.read\_config  
    - Util\_files.createAudFilesInLocalDir  
    - Util\_files.writeScriptsChecksumInLog  
    - Util\_files.getFilesQuantityInRemoteDir  
    - Util\_files.getParamikoSSHCLient  
    - Util\_files.getFilesQuantityInDir  
    - configparser  
    - sys

## Loader\_by\_db.py

A module for organizing audit files (\*.aud ) by database and loading them into the destination directory defined by configuration.

Source and destiny directory are defined in config.properties like next example:

    [LOCAL\_SERVER]  
    # Local directories  
    local\_dir\_audit\_files = /home/arcsight/auditCBS  
    local\_dir\_audit\_files\_by\_db =  /home/arcsight/auditCBS\_Processed

In source directory (local\_dir\_audit\_files) is expected subdirectories like this  
    202405  
        billdb-1-1-m-0  
        billdb-1-1-m-1  
        billsharedb-1-1-m-0  
        billsharedb-1-1-m-1  
        bmpdb-1-1-m-0  
        bmpdb-1-1-m-1  
        edrdb-1-1-m-0  
        edrdb-1-1-m-1  
        meddb-1-1-m-0  
        meddb-1-1-m-1  
        usrdb-1-1-m-0  
        usrdb-1-1-m-1  
        uvcdb-1-1-m-0  
        uvcdb-1-1-m-1

The result in destiny directory (local\_dir\_audit\_files\_by\_db) there are sub-directories named same as databases:  
    billsharedb  
    bmpdb  
    edrdb  
    meddb  
    usrdb  
    uvcdb

When this module is called directly, details of this task are logged into **loader\_by\_db-{logFileDateStr}.log** file  
With logFileDateStr = datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")

### Usage Examples:

    1. Organize audit files by database and load them into the destination directory:  
    >>> from loader\_by\_db import organizeAuditFilesbyDB  
    >>> organizeAuditFilesbyDB("/source/dir", "/destination/dir", "20240504", "subdir1\nsubdir2")  
    'OK'

    2. Run the loader\_by\_db module:  
    >>> from loader\_by\_db import loader\_by\_db  
    >>> loader\_by\_db("20240503")

### Dependencies

    - os  
    - logging  
    - traceback  
    - glob  
    - re  
    - shutil  
    - datetime  
    - Util\_files.read\_config  
    - Util\_files.deleteDirContent  
    - Util\_files.writeScriptsChecksumInLog  
    - Util\_files.getFilesQuantityInDir  
    - sys

## Compress\_aud\_files.py

A module for compressing audit files based on configuration settings into config.properties file.

Source \*.aud files to be compressed is defined by ***local\_dir\_audit\_files*** configuration parameter.

A result of this task is a file named as {month\_str}-aud-files.tar.gz, with mont\_str=%Y%m, for example:  
    202405-aud-files.tar.gz

This compressed file example contains all may/2024 audit files and is saved into same ***local\_dir\_audit\_files*** directory.

When this module is called directly, details of this task are logged into **compress\_aud\_files-{logFileDateStr}.log** file  
With logFileDateStr = datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")

### Usage Examples:

    1. Compress audit files based on default settings:  
    >>> from compress\_aud\_files import compress\_aud\_files  
    >>> compress\_aud\_files()  
    'OK'

    2. Compress audit files for a specific month:  
    >>> compress\_aud\_files("202405")  
    'OK'

### Dependencies:

    - os  
    - logging  
    - datetime  
    - Util\_files.read\_config  
    - Util\_files.writeScriptsChecksumInLog  
    - Util\_files.compressFiles  
    - sys

## Compress\_db\_aud\_files.py

A module for compressing audit files based on configuration settings into config.properties file.

Source \*.aud files to be compressed is defined by ***local\_dir\_audit\_files\_by\_db*** configuration parameter.

A result of this task is a file named as {month\_str}-aud-files.tar.gz, with mont\_str=%Y%m, for example:  
    202405-aud-files.tar.gz

This compressed file example contains all may/2024 audit files and is saved into same ***local\_dir\_audit\_files\_by\_db*** directory.

When this module is called directly, details of this task are logged into **compress\_db-aud\_files-{logFileDateStr}.log** file.  
With logFileDateStr = datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")

### Usage Examples:

    1. Compress audit files based on default settings:  
    >>> from compress\_aud\_files import compress\_aud\_files  
    >>> compress\_aud\_files()  
    'OK'

    2. Compress audit files for a specific month:  
    >>> compress\_aud\_files("202405")  
    'OK'

### Dependencies:

    - os  
    - logging  
    - datetime  
    - Util\_files.read\_config  
    - Util\_files.writeScriptsChecksumInLog  
    - Util\_files.compressFiles  
    - sys

# 7. Technical Support

If you encounter any issues during the installation that ayou can't resolve or have any additional questions, feel free to contact our technical support team at [gonzalesv@domain.com](mailto:gonzalesv@tigo.net.bo)

# 8. Appendix

## Glossary

|  |  |
| --- | --- |
| Term | Definition |
| ETL | Extract Transform and Load |
| Python module | A file with the “. py” extension containing Python code that can be imported inside another Python script |

## Validation modules

1. It is necessary validate Python installed modules, so, execute, using Python, the file Validate\_modules.py
   1. Example 1, for windows: *C:/Users/gonzalesv/AppData/Local/Programs/Python/Python312/****python.exe*** *d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/new-scripts/****Validate\_modules.py.***
   2. Example 2, for linux: ***python3*** *“****/root/Scripts****/****Validate\_modules.py****”.*
2. Result execution for success is (Realize that "NOK modules" section is empty) :
   1. *OK modules:*  
      *1. Yes module named os*  
      *2. Yes module named logging*  
      *3. Yes module named configparser*  
      *4. Yes module named re*  
      *5. Yes module named datetime*  
      *6. Yes module named traceback*  
      *7. Yes module named glob*  
      *8. Yes module named sys*  
      *9. Yes module named shutil*  
      *10. Yes module named hashlib*  
      *11. Yes module named smtplib*  
      *12. Yes module named paramiko*  
      *13. Yes module named tarfile*  
      *14. Yes module named base64*  
      *15. Yes module named from Util\_string import getAuditedActivity, getDataBetween, getObjectNameFromQuery*  
      *16. Yes module named from scp import SCPClient*  
      *17. Yes module named from Util\_files import deleteDirContent, createSSHClient, read\_config, createAudFilesInLocalDir, writeScriptsChecksumInLog, getFilesQuantityInRemoteDir, getParamikoSSHCLient, getFilesQuantityInDir, compressFiles*  
      *18. Yes module named from Extractor import extractor*  
      *19. Yes module named from Loader\_by\_db import loader\_by\_db*  
      *20. Yes module named from Compress\_aud\_files import compress\_aud\_files*  
      *21. Yes module named from os.path import basename*  
      *22. Yes module named from email.mime.application import MIMEApplication*  
      *23. Yes module named from email.mime.multipart import MIMEMultipart*  
      *24. Yes module named from email.mime.text import MIMEText*  
      *25. Yes module named from email.utils import COMMASPACE, formatdate*

***NOK modules****:*

1. If some Python module is missing, then it will be listed in "NOK modules" section, for example
   1. *NOK modules:*  
      *1.* ***No module named*** *'module\_example'*
2. Missing modules must be installed. You could follow official python guide in <https://docs.python.org/3/installing/index.html>
   1. Example: *python -m pip install SomePackage*

## Content example of .aud file

A content example of audit file (.aud) is:

|  |
| --- |
| UTC-4:00 2024-04-17 03:00:02.926  LENGTH: "231"  SESSIONID:[3] "977" STMTID:[1] "0" USER:[4] "USR1" HOST:[11] "11.12.1.123" **ACTION**:[21] "PREP\_EXEC[AUTOCOMMIT]" RETURNCODE:[8] "GS-00000" SQLTEXT:[79] "    truncate table SOME\_TABLE\_BYPASS01    "    UTC-4:00 2024-04-17 03:00:04.278  LENGTH: "228"  SESSIONID:[3] "977" STMTID:[1] "0" USER:[4] "USR1" HOST:[11] "11.12.1.123" **ACTION**:[21] "PREP\_EXEC[AUTOCOMMIT]" RETURNCODE:[8] "GS-00000" SQLTEXT:[76] "  alter table SOME\_TABLE\_RE drop partition \_SYS\_P4460  "    UTC-4:00 2024-04-17 06:33:48.564  LENGTH: "234"  SESSIONID:[2] "75" STMTID:[1] "0" USER:[15] "LONG\_USER\_NAME1" HOST:[9] "127.0.0.1" **ACTION**:[7] "PREPARE" RETURNCODE:[8] "GS-01001" SQLTEXT:[89] "UPDATE SCHEM1.SOME\_TABLE1 SET RECHARGE\_AMT=10000 WHERE RECHARGE\_LOG\_ID=138300010024049163"    UTC-4:00 2024-04-17 06:34:52.057  LENGTH: "1990"  SESSIONID:[2] "75" STMTID:[1] "0" USER:[15] "LONG\_USER\_NAME1" HOST:[9] "127.0.0.1" **ACTION**:[7] "EXECUTE" RETURNCODE:[8] "GS-00000" SQLTEXT:[1843] "INSERT INTO SCHEM1.SOME\_TABLE1 ...    UTC-4:00 2024-04-27 03:00:04.278  LENGTH: "257"  SESSIONID:[3] "977" STMTID:[1] "0" USER:[9] "GONZALESV" HOST:[11] "11.12.1.123" ACTION:[21] "PREP\_EXEC[AUTOCOMMIT]" RETURNCODE:[8] "GS-00000" SQLTEXT:[97] "  **alter** **table** LONG\_SCHEMA\_DWH1.SOME\_TABLE\_OFF1 **drop** partition PRT\_D\_20240222  "    UTC-4:00 2024-04-27 03:00:04.278  LENGTH: "214"  SESSIONID:[3] "977" STMTID:[1] "0" USER:[9] "GONZALESV" HOST:[11] "11.12.1.123" ACTION:[21] "PREP\_EXEC[AUTOCOMMIT]" RETURNCODE:[8] "GS-00000" SQLTEXT:[55] "**truncate** **table** LONG\_SCHEMA\_NA1.LONG\_TABLE\_NAME\_SCHEMAN1" |

Code Block 2 AUD file Content example

## Content example of Activity report

An example of report file with all data from each audited activity is written as result of checkForAuditedActivities execution is:

|  |
| --- |
| 2024-05-10 10:52:35 PM ======================Chesksum script files==========================  2024-05-10 10:52:35 PM 1: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/config.properties - de2fa0d0372816c539ba6fc7d222b848  2024-05-10 10:52:35 PM 2: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Extractor.py - 519a6063c4a335f0f63b8fe906f007bb  2024-05-10 10:52:35 PM 3: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Loader\_by\_db.py - acb0cf0809b6bae4c6c7a878a1a9c0e7  2024-05-10 10:52:35 PM 4: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Activity\_report\_generator.py - c43bebe4db870940ef02975c480b042b  2024-05-10 10:52:35 PM 5: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Compress\_aud\_files.py - 5cc16f8e2f3207b30ec589de9028756a  2024-05-10 10:52:35 PM 6: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Compress\_db\_aud\_files.py - a68cafd72b656d5b1b3241628caa4de1  2024-05-10 10:52:35 PM 7: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Util\_string.py - 653d6089edf7affd6ef726bb889c343d  2024-05-10 10:52:35 PM 8: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Util\_files.py - 9f2ae0732338a218148875557e181b34  2024-05-10 10:52:35 PM ================Start auditing database activities ===================  2024-05-10 10:52:35 PM Script running: Activity\_report\_generator.py  2024-05-10 10:52:35 PM Activity report using audit files from date: 202405  2024-05-10 10:52:35 PM Source directory: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files-by-db/auditCBS\_Processed  2024-05-10 10:52:35 PM Fecha y Hora Usuario de BD Hostname Linea Actividad Schema Table Query Archivo  2024-05-10 10:52:35 PM 2024-04-17 03:00:02.926 ARDB 10.24.4.209 221639 Truncate ARDB AR\_HIS\_BATCH\_BYPASS truncate table AR\_HIS\_BATCH\_BYPASS d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files-by-db/auditCBS\_Processed\billdb\billdb0-zengine\_20240502031134948.aud  2024-05-10 10:52:35 PM 2024-04-17 03:00:04.278 ARDB 10.24.4.209 221867 Drop ARDB AR\_IST\_RECORD alter table AR\_IST\_RECORD drop partition \_SYS\_P4460 d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files-by-db/auditCBS\_Processed\billdb\billdb0-zengine\_20240502031134948.aud  2024-05-10 10:52:35 PM 2024-04-17 03:00:04.407 ARDB 10.24.4.209 221873 Drop ARDB AR\_QUERY\_RECORD alter table AR\_QUERY\_RECORD drop partition \_SYS\_P4460 d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files-by-db/auditCBS\_Processed\billdb\billdb0-zengine\_20240502031134948.aud  2024-05-10 10:52:35 PM 2024-04-17 04:10:03.303 BILLDB 10.24.4.208 37715 Drop BILLDB BB\_LOG\_SPLITFILE alter table BB\_LOG\_SPLITFILE drop partition PART\_20240118 d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files-by-db/auditCBS\_Processed\billdb\billdb0-zengine\_20240502064559403.aud  2024-05-10 10:52:35 PM 2024-04-17 04:10:03.372 BILLDB 10.24.4.208 37721 Drop BILLDB BB\_FILE\_SUMMARY alter table BB\_FILE\_SUMMARY drop partition PART\_20240118 d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files-by-db/auditCBS\_Processed\billdb\billdb0-zengine\_20240502064559403.aud |

Code Block 3 Activity report

## Content example of Extract\_and\_load module

|  |
| --- |
| 2024-05-10 03:12:09 PM Script running: Extract\_and\_load.py  2024-05-10 03:12:09 PM ======================Chesksum script files==========================  2024-05-10 03:12:09 PM 1: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/config.properties - 28339052a674f411fd4335478bb7995a  2024-05-10 03:12:09 PM 2: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Extractor.py - b05d705ffcffb4e384eb2f32bdd2fb6c  2024-05-10 03:12:09 PM 3: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Loader\_by\_db.py - acb0cf0809b6bae4c6c7a878a1a9c0e7  2024-05-10 03:12:09 PM 4: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Activity\_report\_generator.py - d48a5c1eb0a6468c4885f398508c5b9a  2024-05-10 03:12:09 PM 5: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Compress\_aud\_files.py - 5cc16f8e2f3207b30ec589de9028756a  2024-05-10 03:12:09 PM 6: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Compress\_db\_aud\_files.py - a68cafd72b656d5b1b3241628caa4de1  2024-05-10 03:12:09 PM 7: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Util\_string.py - 234ae5465a4168a9c94fabca03e41c49  2024-05-10 03:12:09 PM 8: d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/Util\_files.py - 9f2ae0732338a218148875557e181b34  2024-05-10 03:12:09 PM =========================Start extraction============================  2024-05-10 03:12:09 PM Script running: Extractor.py  2024-05-10 03:12:09 PM Extract files from date: 20240509  2024-05-10 03:12:09 PM Start deleting files from directory d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files/auditCBS/20240509  2024-05-10 03:12:09 PM Total files/directories deleted: 0  2024-05-10 03:12:09 PM Start copying files from server 172.28.10.88, directory /home/mobarmy/auditlogs to d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files/auditCBS/20240509  2024-05-10 03:12:09 PM Connected (version 2.0, client OpenSSH\_7.4)  2024-05-10 03:12:09 PM Authentication (password) successful!  2024-05-10 03:12:09 PM Connected (version 2.0, client OpenSSH\_7.4)  2024-05-10 03:12:10 PM Authentication (password) successful!  2024-05-10 03:12:10 PM The directory d:/projects/67-992-CBS-UPGRADE/06Scripts/desarrollo/**new**-scripts/destiny-audit-files/auditCBS/20240509/billdb-1-1-m-0 created successfully!  2024-05-10 03:12:10 PM [chan 0] Opened sftp connection (server version 3)  2024-05-10 03:12:10 PM [chan 0] sftp session closed.  2024-05-10 03:12:10 PM File quantity in source directory billdb-1-1-m-0: 1  2024-05-10 03:12:12 PM 1: zengine\_20240509034934079.aud  .. |

Code Block 4 Extract and load module log

## References

1. <https://docs.python.org/3/installing/index.html>
2. <https://docs.python.org/3/tutorial/modules.html>