Genesys Care Workbench - Data Purge Utility User Guide

Genesys Care/Support current

1/23/2022

Table of Contents

Genesys Care Workbench 8.5 GDPR - Data Purge Utility User's Guide

Overview

The General Data Protection Regulation (GDPR) is a regulation in EU law on data protection and privacy for all individuals within the European Union and the European Economic Area. There is a requirement for Genesys customers located in Europe or serving the European market to meet the standards of the EU GDPR.

To enable them to meet these requirements, Genesys customers using the Genesys Care Workbench 8.5 need to have the capability to purge any application data that may contain PII (Personally identifiable information). This Workbench 8.5 - Data Purge Utility enables that for all 8.5.x Workbench releases.

New in this Release

This section describes the new functionality for each release.

Release 8.5.100

This is the first public release of the Genesys Care Workbench - Data Purge Utility.

Known Issues and Limitations

There are currently no known issues for this release of the Genesys Care Workbench - Data Purge Utility.

Prerequisites

- Java™ Platform Standard Edition Runtime Environment 8, 64-bit (JRE™ 8)

> **Important**
> The Data Purge Utility Tool must be run on the same host as the Workbench.

How to use the Workbench 8.5 GDPR Data Purge Utility

The Workbench 8.5 GDPR - Data Purge Utility (DPU) is designed to allow deletion of data from the Workbench Cassandra database tables and Workbench log files older than a set number of days. This tool needs to be run from the command line as a standalone java application, and is named **DataPurgeUtility.jar**.

DPU provides two functions in one tool:

1. To delete any records older than 30 days, from all main Cassandra DB tables, and
2. To delete all Workbench log files older than 30 days.

> **Important**
> Once the data is deleted using DPU, it is unrecoverable. Workbench 8.5 will then display reduced data based on the data that are purged.

Steps to follow when running Data Purge Utility:

1. Stop Workbench application and confirm that all the processes on the Workbench server have terminated. This can be confirmed using the Task Manager on Windows and using `ps -aelf | grep java` and ensuring that all the java processes for karaf and Cassandra are no longer seen.

   If they still persist, manually kill them using the following instructions:

   a. Windows:
      i. Start Task Manager
      ii. Locate the java process for Cassandra or karaf
      iii. Right click and select **End Process**

   d. Linux:
      i. Run `ps -aelf | grep java`
      ii. Identify the process ID for karaf or Cassandra and then execute: `kill -9 <process id>`

2. Start Cassandra:

   a. Windows:
      i. Create a command prompt at <WORKBENCH_INSTALLATION_FOLDER>

ii. Run: `cassandra\bin\cassandra.bat`

c. Linux:

i. cd to <WORKBENCH_INSTALLATION_FOLDER>

ii. run: Cassandra/bin/Cassandra

2. Run Data Purge Utility and wait until it completes its execution.

   Below is an example of the syntax to be used. This command should be run from a command prompt in the directory of the DPU tool:
   ```
   java -jar DataPurgeUtility.jar -h 127.0.0.1 -p 9042 -d 30 -f table_specs.txt
   -log_path "<WORKBENCH_INSTALLATION_DIRECTORY>" -cassandra_bin_path c:\GCTI\
   WB_server\cassandra\bin -perform_compaction -cassandra_jmx_port 17199
   ```
   This command deletes all database records and WorkBench log files older than 30 days and performs database compaction.

3. Stop Cassandra manually by following 1.a or 1.b above.
4. Start Workbench.

Following are descriptions of the parameters for this utility.

Mandatory Parameters

1. `-h <ip>:` It specifies the IP address of where the Cassandra Database server is running. Typically, the same as the Workbench host.
2. `-p <port>:` It specifies the port used by the Cassandra Database server is running. This is the value of the Database Transport Port set during Workbench installation.
3. `–d <# of days>:` It specifies the retention period in number of days. Table records older than the specified number of days will be deleted.

> **Important**
>
> In order to meet GDPR requirements, we recommend running the **Genesys Care Workbench - Data Purge Utility** regularly to ensure no more than 30 days of data is kept. This can be done by using a combination of how frequently you execute the tool and the retention period set with the -d option. For example setting **-d 20** requires to run the tool every 10 days.

Cassandra Database Table Deletion Parameters

4. In order to examine and remove the Cassandra Table Data, follow one of the two following sets of specifications:

   a. This set can be used for an individual table i.e. if data from only one table needs to be examined.

   i. `-t <tablename>:` The name of the table from which data will be removed if it satisfies the

criteria

ii. `–c <columnname>:` The name of the column that holds the timestamp.

iii. `–a <additional key names>:` The name(s) of other columns that are part of the primary key until the column that holds timestamp.

iv. `–pk<0 or 1>:` Whether or not the column that holds timestamp is part of the partition key. 0 indicates *No* and 1 indicates *Yes*.

v. `–prk<0 or 1>:` Whether or not the column that holds timestamp is a part of the Primary key. 0 indicates *No* and 1 indicates *Yes*.

f. If there are more tables that need their data examined and removed if needed, then the following specification is used:

i. `–f <filename>:` Full path of the file that contains one line per table. Each line contains the five parameters explained above for each table. Please note that if this option is specified, the five specifications in 4.a. above will be ignored even if additionally specified. (See File Format section below for details.)

2. `-set_ttl:` If present, the **Time-To-Live** property (in seconds) corresponding to the specified number of days is set for the tables.

   For Example: To perform Cassandra compactions and set ttl (time-to_live)
   ```
   java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -cassandra_bin_path
   c:\GCTI\WB_server\cassandra\bin -perform_compaction -cassandra_jmx_port 17199
   -set_ttl
   ```

3. `-tombstone_seconds <integer value>:` This is an optional parameter that contains the value of **gc_grace_seconds** Cassandra Table property. This property will be temporarily set to the specified value and later restored to the default value of **8640000** after the deletion of data. This option can be used if an error is encountered during the data deletion and will help to work around the failure(s) due to greater than the maximum allowed tombstone entries. See Troubleshooting for more details.

Cassandra Compaction Parameters

The usage of these compaction options is recommended to reduce the disk usage of Cassandra but is not mandatory.

4. `-cassandra_bin_path <full path>:` This specifies the full path for the bin where the Cassandra utilities are located.
5. `-perform_compaction:` If present, **Nodetool Compact** from the Cassandra bin folder will be executed at the end of the operation.
6. `-cassandra_jmx_port:` JMX Port that was used when installing Workbench. If not specified, default value of 7199 will be used when invoking **Nodetool** command.

> **Tip**
> It is highly recommended to run the above compaction parameters after any deletion of the records from the Cassandra database to optimize the memory usage.

Log File Deletion Parameters

7. –`log_path <full path>:` This is an optional parameter that contains the full path (not relative) to a folder. Any file in that folder that has a **last modified timestamp** older than the specified number of days will be removed. This option can be used to remove Workbench logs older that the specified number of days.
8. `-recursive:` While deleting the log files, the utility would look for log files in sub-directories if this flag is specified.
9. `-log_file_extensions <extensions>:` These are comma separated file extensions that are considered to be log files. While deleting these log files, the utility will look files with these extensions and other rotated files and delete them only if they are older than the specified number of days.

   For Example: `-log_file_extensions log` – This will look for all *.log, *.log.<count> files for deletion. The *.log.<count> files are rotated files (Example: def.log.9).

Command Line Examples

Following are some examples of command line.

Example 1: To remove entries from just one table

```
  java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -t
context_tree_new.events_by_type_name -c ts
  -d 30 -a event_type event_name -pk 0 –prk 1
```

This command line example applies to a table definition as described below..

```
CREATE TABLE context_tree_new.events_by_type_name (<br>
    event_type int,
    event_name text,
    ts timestamp,
    app_type text,
    event_id uuid,
    eventdetails blob,
    hourkey text,
    level int,
    msg text,
    seconds bigint,
    sent boolean,
    status boolean,
    PRIMARY KEY ((event_type, event_name), ts)<br>
);
```

Example 2: To remove entries from a set of tables, where they entries are stored in a file *table_specs.txt*

```
  java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -d 30 -f C:\\table_specs.txt
```

Example 3: To remove log files from the disk

```
 java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -log_path c:\logs\wb_server\
 -recursive -log_file_extensions log
```

Example 4: To perform Cassandra compactions

```
 java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -cassandra_bin_path
 c:\GCTI\WB_server\cassandra\
 bin -perform_compaction -cassandra_jmx_port 17199
```

Example 5: To perform Cassandra compactions and set Time-To-Live (ttl)

```
java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -cassandra_bin_path
c:\GCTI\WB_server\cassandra\
bin -perform_compaction -cassandra_jmx_port 17199 -set_ttl
```

File Format and Examples

If a file is used (-f option) to specify the tables to purge, the format of the file specified in the above point 4-b is as follows:

Each line in the text file contains five fields that are TAB separated.

The fields are:

```
tablename<TAB>columnname<TAB>additional key names<TAB>pk value<TAB>prk value
```

The following are the definitions of these fields.

1. **tablename :** The name of the table from which data will be removed if it satisfies the criteria.
2. **columnname :** The name of the column that holds the timestamp.
3. **additional key names :** The name(s) of other columns that are part of the primary key until the column that holds timestamp. Each column name should be comma separated.
4. **pk value :** Whether or not the column that holds timestamp is part of the partition key. 0 indicates no and 1 indicates yes.
5. **prk value :** Whether or not the column that holds timestamp is a part of the Primary key. 0 indicates no and 1 indicates yes.

> **Important**
>
> A file *table_specs.txt* is included with the JAR that contains specifications for all the Workbench tables. This can be used to remove data from all Workbench tables.

Some Examples

Example 1

Consider the following table definition:

```
CREATE TABLE act.alerts_triggered_by_call
         call_id uuid,
         alert_id uuid,
         alert_condition text,
         alert_timestamp timestamp,
         alert_type text,
         parameters map<text, text>,
         PRIMARY KEY (call_id, alert_id)
) WITH CLUSTERING ORDER BY (alert_id ASC)
```

The entry in the file for this would be:

```
   act.alerts_triggered_by_call            alert_timestamp          call_id, alert_id
no        no
```

Since *alert_timestamp* is not part of the partition key, it is not a part of the primary key.

A sample of the command for this table would be:

```
   java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -d 30 -t
act.alerts_triggered_by_call -c
   alert_timestamp -a "call_id,alert_id" -pk 0 -prk 0
```

Example 2

Consider the following table:

```
   CREATE TABLE beholder.changes (
          timemodified timestamp,
          dbid int,
          changekey text,
          changeby text,
          changetopobjectname text,
          changevalue text,
          objecthostip text,
          objecthostname text,
         operationtype text,
         PRIMARY KEY (timemodified, dbid, changekey)
   ) WITH CLUSTERING ORDER BY (dbid ASC, changekey ASC)
```

The entry in the file for this would be:

```
   beholder.changes              timemodified            dbid,changekey          yes
yes
```

Since *timemodified* is a part of the partition key and it is a part of the primary key.

A sample of the command for this table would be:

```
   java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -d 30 -t beholder.changes -c
timemodified -a "dbid,changekey" -pk 1 -prk 1
```

Troubleshooting

Suggested workarounds in case of errors in purging data

The Data Purge Utility may encounter errors while attempting to purge data from the specified tables. The following workarounds can be used to help overcome these errors:

1. Modification of Cassandra configuration file:
    a. Login to Genesys Administrator and stop Workbench, in order to stop the Cassandra process.
    b. In an OS command prompt window on the Workbench host, change directory to *<WORKBENCH_INSTALLATION>/cassandra/conf>*
        i. In this directory, copy file *cassandra.yaml* to *cassandra.yaml.sav*
    b. Open file *cassandra.yaml* in a text editor and change the following:

cassandra.yaml	Current value	New Value
read_request_timeout_in_ms	5000	30000
request_timeout_in_ms	10000	60000

    c. Using Genesys Administrator, start Genesys Workbench, in order to restart the Cassandra process.
    d. Re-run the DataPurgeUtility tool.

> **Important**
> If there are a lot of records in the database, it is still possible to see the **Read timeout** errors when deleting records. In that case, it is advisable to change both the timeout numbers to 3600000 instead of 30000 and 60000 (refer the above table).

    e. After purging, do the following to roll back the Cassandra configuration changes,
        i. Stop Workbench/Cassandra (see step a.)
        ii. Copy *cassandra.yaml.sav* to *cassandra.yaml*.
        iii. Start Workbench/Cassandra (see step d.).

3. Run the DataPurgeUtility using a parameter `-tombstone_seconds` when invoking DataPurgeUtility. Specify a small integer value **e.g. 300** (for 5 minute). Please note that the **GC_GRACE_SECONDS** value for a specific table in Cassandra is temporarily set to this specified value. The value is restored to the default value of **8640000** (10 days) after the data is deleted. An example of command line is given below.

```
java -jar DataPurgeUtility.jar -h 135.17.180.184 -p 9042 -t
```

```
context_tree_new.events_by_type_name -c ts
-d 30 -a event_type event_name -pk 0 –prk 1 -tombstone_seconds 300
```