# GDPR Scheduled Cleaner

## Defining the SObject fields to be cleaned

The two ways that I considered were either having the fields defined as a Set<String> constant, which would have been nicer from a coding point of view, however to make it easier for admins to handle quicker without the need to get a developer involved, I chose to use Custom Metadata. This also means that a full development process of deploying a new version of the class doesn't need to be followed through, and it can be implemented faster with the ability to do quicker testing in sandboxes.

## Apex Classes

| | |
|---|---|
| **BatchGDPRCleaner.cls** | Main batch job to handing selecting the right Case records and related Account and Contact records, and cleaning of the data. The close date to use as a filter has a default value set to greater than a year from the day the batch job is run. This can be changed when running the job in the developer console if the batch job needs to be run manually.<br><br>Test cover: 100% |
| **BatchGDPRCleanerTest.cls** | Test class for BatchGDPRCleaner.cls |
| **ScheduleGDPRCleaner.cls** | The scheduler for the batch job. Currently set up to run every day at 02:00.<br>The job name is: GDPR Daily Cleaner<br>The CRON time is set in a String variable without the final attribute, so it can be set manually if desired, otherwise running the following line will set up the scheduled job to run the batch every day at 2am:<br>ScheduleGDPRCleaner.schedule();<br><br>Test cover: 100% |
| **ScheduleGDPRCleanerTest.cls** | Test class for ScheduleGDPRCleaner.cls |
| **UtilityGDPRCleaner.cls** | The Apex class which contains all needed logic referenced in the batch job.<br><br>Test cover: 97% |
| **UtilityGDPRCleanerTest.cls** | Test class for UtilityGDPRCleaner.cls |

## Custom SObjects

### Debug Log

Contains various fields than can be used to record data specific to errors that happen in any Apex code. Allows for tracking class name, exceptions, pinpointing line numbers, and additional info.

## Custom Metadata

- GDPR_Account_Field__mdt
- GDPR_Case_Field__mdt
- GDPR_Contact_Field__mdt

All three custom metadata objects have the same fields available to them. These could have been combined into one custom metadata object with another picklist to select the SObject type, however for this I felt that this would make it more visible that

### Field Name – String

Length 50. I figured this would be long enough to capture most API names for fields. Can easily be extended if need be. This would benefit from a validation rule or trigger to make sure the field exists on the jobs.

### Field Type – Picklist

Three options are available:

- Lookup
- Text
- Number

These three options determine the behaviour of the cleaning for the batch job.

| | |
|---|---|
| **Lookup** | Changes the lookup value to null |
| **Text** | Uses the UtilityGDPRCleaner function randString() to create a randomly generated String of length 25. |
| **Number** | Uses the UtilityGDPRCleaner function randNumber() to create a randomly generated number with a maximum of 10 digits on the left of the decimal point. |

## Data Cleaning Process

The data cleaning process starts with selecting all Case records which meet the close day and non-processed criteria, then related Account and Contact records are queried.

Any lookup fields are changed to a null value.

Any text fields are changed to a random string of length 25 made up of numbers, uppercase, and lowercase characters.

Any numeric fields are changed to a randomly generated Decimal with a maximum of 10 digits to the left of the decimal point. This will allow for different Decimal lengths to be cleaned with the function and Salesforce cutting the number down to the correct size.

Each group of records is updated separately. Any errors are recorded n a record of type Debug_Log__c. This is done on a record-by-record basis. There are a few areas where errors are caught.

The AllOrNothing flag is used to capture errors and report on them.

For testing purposes a new field was introduced for a filter:

Case.TestClosedDate__c

This needs to be set to a date greater than a year old for the filter to pick up the correct case records.

## Assumptions

- Field history is not enabled for the fields that are to be cleaned.
- It should be easy to update and change the affected fields with config, as fields are added with config.

## Nice to haves

- As a final part of the Batch job it would be a nice option to include an email alert to an email list to notify of Debug_Log__c records being created.
- The “cleanData*()” functions can be refactored into one for SObjects.
- Better cleaning of records when an error is raised for less human interaction. E.g. With the current functionality if an Account or a Contact fails the cleaning process but the case passes, it will be flagged in the debug log, but it won’t be picked up again the next time the job runs.
- Could be useful to introduce anonymisation of birthdays.
- Custom metadata field “Field Name” could be removed and the label used in its place.
- Better handling of shorter strings when desired text strings are have a shorter length than the current 25 length randomly generated string.