Customizing GDPR in netFORUM Enterprise

This document explains opportunities to personalize the GDPR features of netFORUM Enterprise to your organization's specific needs. Each organization may approach GDPR compliance differently, so these general instructions are a starting point for your own organization's standard operation procedures.

The screen shots in this document show netFORUM version 2017. The features function similarly in earlier versions of netFORUM, but the user interface differs slightly.

Contents

Customizing GDPR Features

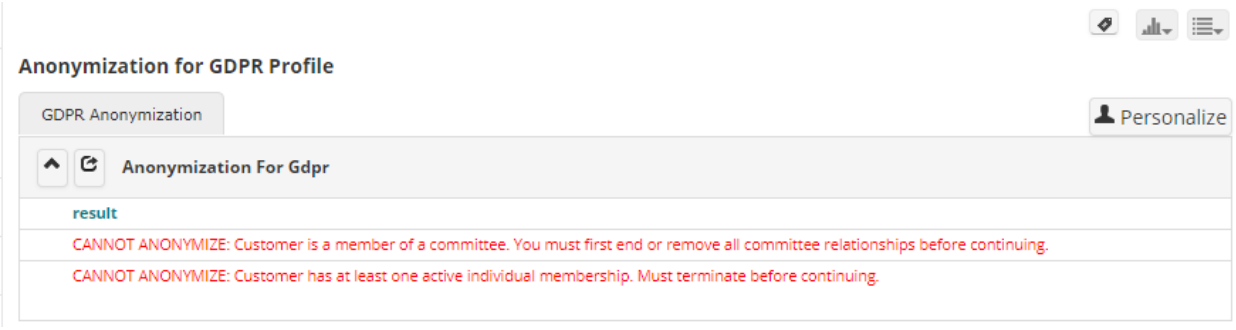
If your organization desires to add or change validations, which personally identifiable information (PII) data is included in the anonymization operation, or if you'd like to add additional pieces of netFORUM to the Data Export, this document explains how you can do this.

In order to customize the anonymization operation, you will need to run Microsoft SQL Server Management Studio and you must have Transact SQL skills in order to write basic UPDATE and SELECT commands. You must also know the netFORUM data model. You will not be able to change this from the netFORUM user interface. In order to include more data elements in the Data Export, you will need to operate the netFORUM Toolkit.

Validations

When a user runs the anonymization operation, netFORUM first performs a series of 20+ distinct validations to check to see if the individuals has any active business with the organization. If the individual does, then the anonymization operation will stop without updating any data, and show the user a message for each validation that failed.

Here is an example of what a user will see after they attempt to anonymize an individual who has active business:



Validation failures during the anonymization operation

All the validations will be tested, and any that fail will be listed, as shown above.

If all the validations succeed, then netFORUM will proceed to execute the anonymization operation.

Configuring Validations

Validation business rules are managed in a database table called co_individual_gdpr_validation. The initial release of GDPR contains more than 20 different validations. Each row in this table executes a SQL select command that returns a recordset with 0 or more rows, which looks for the existence of data, linked to the individual, in a table that would be considered “active business”. If one or more rows are returned, then the validation fails. If 0 rows are returned, then the validation passes.

For example, the following command looks for the existence of an invoice, associated to the individual, in a non-closed accounting batch:

```
SELECT 1 FROM ac_invoice WHERE inv_delete_flag = 0 AND (inv_cst_key = @key OR
inv_cst_billing_key = @key OR inv_ind_cst_billing_key = @key) AND inv_bat_close_flag = 0
```

The @key parameter in the statement will be substituted at runtime with Customer Key of the individual you are attempting to anonymize.

If this statement returns one or more rows, then the validation fails. The anonymization operation cannot continue until the active business is resolved. The resolution could be different for each type of data.

For example, one type of active business is having an active committee assignment:

```
SELECT 1 FROM mb_committee_x_customer WHERE cmc_cst_key = @key AND (cmc_end_date IS NULL
OR cmc_end_date > GETDATE())
```

To resolve this validation, a user must enddate all the individual's committee assignments.

Adding a New Validation

If you wish to include additional validations, you can add one in the following way.

First, in SQL Server Management Studio, write a SQL select command that looks for the existence of data belonging to an individual, in a particular table. Here is a starting point:

```
declare @key uniqueidentifier = 'AC87F1D1-AF3B-4BF1-A2B9-2F1B675642B4'

select 1 from co_customer_alias where cal_cst_key=@key
```

For the @key variable on the first line, set to the value to be that of a specific Customer Key where that customer does exist in the table. Make sure the SELECT command returns a row. Once you have the SQL SELECT command working successfully, write a INSERT command as shown below to insert a new row into the co_individual_gdpr_validation table like this:

```
insert co_individual_gdpr_validation (isActive, query, msg)

values (1, 'select 1 from co_customer_alias where cal_cst_key=@key', 'Customer has a Name
Alias. You must remove all Name Aliases for the customer before continuing.')
```

For the value that goes into the "query" column, do not include the "declare" line in the fragment above; this is just to help test the validity of your SQL command. Include only the SELECT line. Be sure to include the @key variable in the SQL command. The value

for “msg” should be a human readable error message that describes the validation error and how to resolve it.

If the logic for a validation is complex, you can write a stored procedure or UDF and have the stored procedure or UDF execution be in the “query” column.

After running this INSERT command, you should see the new record if you select all the rows from the table:

```
select * from co_individual_gdpr_validation
```

Results | Messages

	number	isActive	query	msg
19	19	1	SELECT 1 FROM mb_membership_proxy WHERE mpr_delet...	Customer has at least one inherited membership. Mu...
20	20	1	SELECT 1 FROM co_customer_x_address JOIN co_address ...	Customer has an address linked to another's addres...
21	21	1	SELECT 1 FROM co_customer_x_phone JOIN co_phone O...	Customer has a phone linked to another's address. ...
22	22	1	SELECT 1 FROM co_customer_x_fax JOIN co_fax ON fax_k...	Customer has a fax linked to another's address. Plea...
23	23	1	SELECT * FROM oe_fulfillment LEFT JOIN ac_invoice_detail ...	Customer has fulfillments that have not been shippe...
24	24	1	select 1 from co_customer_alias where cal_cst_key=@key	Customer has a Name Alias. You must remove all Na...

Results of query

Removing a Validation

If you wish to remove a validation, then update the isActive column for the particular row to 0. We caution that doing so could result in problems if the data is anonymized. You will need to update the specific row by the “number” property which is the primary key of this table:

Sample SQL Command:

```
update co_individual_gdpr_validation set isActive=0 where number=15;
```

If you added a custom validation and you wish to delete it, you can run a delete statement by referencing the number value:

```
delete co_individual_gdpr_validation where number=999;
```

Updating a Validation

If you wish to update a particular baseline validation to change the logic in some way, then we recommend that you make the baseline validation inactive, and then add a new one containing the specific logic that you desire.

Alternately, leave the baseline validation intact and add a new additional one (unless the baseline validation conflicts with your new validation).

Customizing the Anonymization Data

If you wish to alter which personally identifiable information (PII) data netFORUM anonymizes, you may manage those operations by updating or editing the data in the database table co_individual_gdpr_action. In order to do this, you will need to know the netFORUM data model and some basic Transact SQL skills to write UPDATE SQL commands. You must ensure that your SQL command is correct and updates ONLY the specific individual and not all individuals.

The rows in the co_individual_gdpr_action table determine which netFORUM database tables will be anonymized or updated in various way. The default set of operations contains more than 70 different operations against various netFORUM tables, each of which anonymizes specific PII data in netFORUM.

The anonymization operation executes each command, one after another. If any command fails, then the operation will skip to the next one and keep going.

For example, the row that updates phone numbers contains this statement in the “query” column:

```
UPDATE co_phone SET
    phn_number = @gdpr,
    phn_number_display = @gdpr
WHERE phn_cst_key_owner = @key
```

The “msg” column contains the name of the table (co_phone).

As you can see in this command, the purpose is to anonymize the phone number and phone number display columns.

Note the two special parameter values that netFORUM will parse at runtime when the anonymization operation executes. Any commands you run must include these special parameters.

- @gdpr – this will contain a system-generated string that will replace PII data with an anonymized string, for example “GDPR-907987” for an individual whose record number is 907987.
- @key – this will contain the value of the customer key of the customer being anonymized.

You might also wish to NULL certain data fields. Here is an example of the “query” command for the ac_customer_payment_info table, which stores saved payment methods such as electronic payment gateway tokens. This command anonymizes some columns and sets other columns to NULL:

```
UPDATE ac_customer_payment_info SET
    cpi_cc_cardholder_name = @gdpr,
    cpi_cc_number = NULL,
    cpi_cc_number_display = NULL,
    cpi_city = @gdpr,
    cpi_eft_account_number = NULL,
    cpi_eft_account_number_display = NULL,
    cpi_eft_routing_number = NULL,
    cpi_name_on_check = @gdpr,
    cpi_state = @gdpr,
    cpi_street = @gdpr,
    cpi_vault_account = NULL,
    cpi_zip = NULL
WHERE cpi_cst_key = @key
```

Adding an Anonymization Table

You may also update a particular command, or add a new one.

If you wish to update a baseline operation, then we recommend setting the baseline operation isActive column to 0, and adding a new command in a new row to the co_individual_gdpr_action table.

Suppose you have a custom database table containing PII that you would like to anonymize. Suppose the table is called client_xyz_nicknames, with the column n09_nickname containing PII and the column n09_cst_key being the foreign key column containing the customer key. In that case, you would insert the following row into the co_individual_gdpr_action table:

```
Insert co_individual_gdpr_action (isActive, query, msg)

Values (1, 'update client_xyz_nicknames set n09_nickname=@gdpr where n09_cst_key=@key',
'client_xyz_nicknames')
```

After running this command, a new row will exist as shown. The “number” column is an integer identity column that gets set automatically:

```
select * from co_individual_gdpr_action
```

Results | Messages

	number	isActive	query	msg
65	65	1	UPDATE am_surveyor SET svr_surveyor_name = ...	am_surveyor
66	66	1	UPDATE fw_change_log SET log_value = @gdpr, ...	fw_change_log for am_surveyor
67	67	1	UPDATE ex_booth_personnel SET bpr_badge_na...	ex_booth_personnel
68	68	1	UPDATE fw_change_log SET log_value = @gdpr, ...	fw_change_log for ex_booth_personnel
69	69	1	UPDATE ex_exhibitor SET exh_directory_name = ...	ex_exhibitor
70	70	1	UPDATE fw_change_log SET log_value = @gdpr, ...	fw_change_log for ex_exhibitor
71	71	1	update client_xyz_nicknames set n09_nickname=@gd...	client_xyz_nicknames

Another common scenario is if you have PII data in an extender table. For example, suppose you have an individual extender column co_individual.ind_jobboard_username_ext that you wish to anonymize. In this case, add the following row:

```
Insert co_individual_gdpr_action (isActive, query, msg)

Values (1, 'update co_individual_ext set ind_jobboard_username_ext=@gdpr where
ind_cst_key_ext=@key', 'co_individual_ext')
```

If you add a new table, you should also add another entry to anonymize any change log rows related to the record being anonymized. For example, following the phone update

(shown above) a second command runs that anonymizes values in related change log rows:

```
UPDATE fw_change_log SET
    log_value = @gdpr,
    log_value_new = @gdpr
WHERE
    log_mdc_table_name = 'co_phone'
    AND EXISTS (SELECT 1 FROM co_phone WHERE phn_key = log_record_key AND
phn_cst_key_owner = @key)
```

Disabling an Anonymization Table

If you wish to turn off a particular operation, update the isActive column from 1 to 0 as shown. You will need to determine the “number” value which is the primary key column of this table:

```
update co_individual_gdpr_action set isActive=0 where number=71
```

Deleting an Anonymization Table

If you need to delete a row, then execute this command for the particular row you wish to delete:

```
delete from co_individual_gdpr_action where number=71
```

As described earlier, as an alternate to deleting a row, you can set it to be inactive so it will not execute during the anonymization operation:

```
update co_individual_gdpr_action set isActive=0 where number=71
```

We do not recommend deleting baseline rows in this table.

GDPR Data Child Forms

If you wish to add more Child Forms for more data tables that can be exported, you can do so in the Toolkit by adding additional Child Forms and Profile Details, following the same pattern as the baseline child forms.