



KNOWLEDGE SERIES

June 2023 | Volume 77

Finacle Support brings you this fortnightly knowledge bulletin to augment your problem-solving capability. Every edition is put together with utmost diligence to ensure that best practices and known resolutions are shared. In this edition, you will find the following articles:

- **Masking of Account Number in Finacle Online Banking Application**
- **ECUSTBJ Framework - NBP for Custom Batch Jobs**
- **Troubleshoot Queries Using Finacle Support Center**
- **What's New in FSC**

So, let's start reading!

Masking of Account Number in Finacle Online Banking Application

Product: Finacle Online Banking Version: 11.2.x

Due to security concerns and regulatory requirements, banks may need to mask account numbers and card numbers in online banking application. The steps indicated below help achieve the same:

- Setup the below properties in **PROPERTYMANAGER**. The bank admin can navigate to **General Administration: Property Manager → Property Manager Maintenance**
- Set **IS_ACCOUNT_NUMBER_MASKING_REQUIRED = Y**
- Masking patterns can be setup for different account types as shown below. The value of masking pattern can be configured as per the bank's requirement

PROPERTY_NAME	PROPERTY_VAL	PROPERTYDESCRIPTION
IS_ACCOUNT_NUMBER_MASKING_REQUIRED	Y	Indicates whether Account Number Masking required or not
MASKING_PATTERN_ACCT	REDUCE_FIRST_0_LAST_4	Masking Pattern for all other type of Accounts
MASKING_PATTERN_CARD	LEAVE_FIRST_4_LAST_4	Masking Pattern for Credit/Debit Card
MASKING_PATTERN_DEP	REDUCE_FIRST_0_LAST_4	Masking Pattern for Deposit Accounts
MASKING_PATTERN_LON	REDUCE_FIRST_0_LAST_4	Masking Pattern for Loan Accounts
MASKING_PATTERN_OPR	LEAVE_FIRST_4_LAST_4	Masking Pattern for Operative Accounts including Demat

- The below table explains how masking works with different masking patterns using a dummy account number:

Sample Account Number	Masing Pattern	Masked Account Number	Remarks
1234567890123456	REDUCE_FIRST_0_LAST_4	X3456	The last 4 digits from the account number stay as it is. The remaining digits are replaced with a single X.
1234567890123456	REDUCE_FIRST_4_LAST_4	1234X3456	The first 4 and last 4 digits from the account number stay as it is. The remaining digits are replaced with a single X.
1234567890123456	LEAVE_FIRST_0_LAST_4	XXXXXXXXXXXX3456	The last 4 digits from the account number stay as it is. Each one of the remaining digits get replaced by X.
1234567890123456	LEAVE_FIRST_4_LAST_4	1234XXXXXXXX3456	The first 4 and last 4 digits from the account number stay as it is. Each one of the remaining digits get replaced by X.

Using *REDUCE*, the digits identified for masking get replaced by a single X cumulatively.

Using *LEAVE*, each one of the digits identified for masking get replaced by X.

- To show the masked account number in front-end, configure the **"fldFormatter"** value in respective **JSP** form field as **"AccountNumberMaskingFormatter"** (fldFormatter=**"AccountNumberMaskingFormatter"**).
- In case of custom masking implementation, the below parameter available in **"data\EBAppConfig.xml"** can be referred. The helper file configured here can be customized as per business needs

```
<Param name = "ACCOUNT_DISPLAY_NAME_HELPER_ACT" value
="com.infosys.ebanking.common.AccountDisplayNameHelper"/>
```

ECUSTBJ Framework – Non Business Parallelization for Custom Batch Jobs

Product: Finacle Core Banking Version: 10.2.09 and above

The batch jobs implemented using Finacle scripts do not support parallelization by default. During the batch execution of the script, a single process does the selection, execution, and reporting in a sequential manner. This takes longer time to process when the number of records is more.

Finacle provides **Non-Business Parallelization (NBP)** for custom batch jobs through the **ECUSTBJ** framework. This framework provides a structured approach of developing a custom job, by having multiple script hooks for various logical units. It allows breaking down a batch into different logical units, and a single job will be split into multiple script hooks as listed below:

- `<JobName>_SetRegisterInfo.scr` (Mandatory) to set field names and driver register info
- `<JobName>_ListGet.scr` (Mandatory) to fetch records to be processed and put in driver LL
- `<JobName>_RecordProcess.scr` (Mandatory) has the batch processing logic
- `<JobName>_Initialize.scr` to initialize variables etc. in each sub-process
- `<JobName>_NoRecInList.scr` called if no records fetched/put in driver LL by `<>_ListGet.scr`
- `<JobName>_BeforeCommit.scr` to do processing before DB commit is issued by driver
- `<JobName>_AfterCommit.scr` to do processing after DB commit is issued by driver
- `<JobName>_EndOfProcessing.scr` to do processing, at the end of each sub-process
- `<JobName>_PostProcessing.scr` to do processing, once all records have been processed
- `<JobName>_ReposDetails.scr` to set values for repository variables in success/fail MRTs

NBP can be enabled for the custom job using the below parameters:

- **<jobName>_NUM_RECCOUNT** - The variable decides how many records will be processed in each sub-process
- **<jobName>_NUM_JOB** - This variable decides how many processes to be invoked in parallel at a given time.
- **<jobName>_<contextSolId>_NUM_RECCOUNT** - Based on the size of the sol(s), if different configuration is required for how many records should be processed in each sub-process then this sol specific env variable should be set
- **<jobName>_<contextSolId>_NUM_JOB** - Based on the size of the sol(s), if different configuration is required for how many processes should be invoked in parallel at a given time then this sol specific env variable should be set

To enable a job under the custom batch driver framework, a record need to be inserted into **tbaadm.CBJRT** table. The batch job can be configured using **HBJSTM** menu, by specifying the **"File Name/SrInum"** as **"ECUSTBJ/1"**. The execution of the job is done through **ECUSTBJ** menu or **HBJE** menu.

Sample scripts are available in the Product Scripts path (**\$TBA_PROD_ROOT/prodbase/INFENG/scripts**) from version **10.2.15** onwards. We can search for **DMCIF*.scr**.



Troubleshoot Queries with Finacle Support Center!

Household Creation and Mapping in CRM
 Module: *Finacle Core Banking Version: 10.x*

Household in CRM refers to linking or tying Retail CIFs together that belong to a group. And that group can be identified with a unique identifier (Household ID). This information can be used for various bank activities related to the different household accounts...

[Click here](#) to read the full Troubleshooting Document on FSC!

Would you like a particular topic to be discussed in the Support Connect newsletter? Write to us and let us know.

What's New in FSC!

905 incident resolutions and 5 learning videos have been uploaded to FSC.

Bank Holidays
Bank Repository
Bank User approval policy in FEBA
Budget Planner
Latest Image Date in Docker Area

[Click here](#) to visit FSC and view the artifacts.

Do you have the most useful Finacle URLs handy? Bookmark these Finacle links:

- **Finacle Support Center:** <https://support.finacle.com/>
- **Finacle Knowledge Center:** <https://content.finacle.com>
- **TechOnline:** <https://interface.infosys.com/TechonlineV2/base/globallogin>

Hope you like this edition. Is there anything that you'd like to see in the forthcoming series? We'd love to hear from you! Write to us at finaclesupport@edgeverve.com



Better Inspires Better
www.finacle.com