Infosys | Finacle

FINACLE SUPPORT CONNECT

Knowledge Series

July 2021 | Volume 30

Finacle Support brings you this fortnightly knowledge bulletin to augment your problem-solving capability. There is more to it. 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:

- **Significance of the NO_OF_FILES_TO_CACHE Variable**
- **Feature to Print the User Password Automatically and Deliver to the User**
- **Difference Between TBA_DEBUG_LEVEL and TBA_DEBUG Parameters**

So let's start reading!

Significance of the NO_OF_FILES_TO_CACHE Variable

Product: Finacle Core Banking (Customization) Version: 10.x onwards

The **NO_OF_FILES_TO_CACHE** variable is used to identify the number of script files to be cached in an execution. Its value is set to 100 in the product.

During the scenario where one script is called from another and so on without exiting from the parent scripts and thus creating a loop/hierarchy of scripts during execution. Once reaching the limit of 100 scripts, the initial script that has been cached will be deleted and script engine will not be able to trace back to the caller script and will result into unpredictable fatal errors.

How to identify if this limit has exceeded?

1. Caching of the scripts happens for a single LISRVR process
2. Enable a dummy file **sedebug** in the **CDCI_LOGS** user directory, so that system will generate a file **Debug_<pid>.txt** in the user directory during the script execution
3. The content **[iCount >= iNumFilesToCache]** in this debug file signifies that the script files cached has reached the maximum limit

How to avoid the problem of limit being exceeded?

Decrease the number of script calls to as minimum as possible.

Feature to Print the User Password Automatically and Deliver to the User

Product: Finacle Online Banking Version: 11.2.x

The product has a feature to print the user password automatically. Once the admin sets the password of a user in Finacle Online Banking using the **Set Password** functionality in admin application, the bank must run **PWDPRINTBATCH** to print the passwords physically.

Alternatively, the password can be generated and sent immediately via email to the registered user email ID in the **CUSR** table. This is achieved using a **PRPM** parameter **PASSWORD_ALERT_TRIGGER.**

On setting the password of a user, using **Set Password** functionality in the admin application -

- If the **PASSWORD_ALERT_TRIGGER** Property is **N**, on executing **PWDPRINTBATCH** batch it will pick the eligible records to process and print the passwords
- If the **PASSWORD_ALERT_TRIGGER** property is **Y**, then password file in PDF format will be directly generated, without executing PWDPRINTBATCH and sent to the user
- If the email ID is not available, application expects the admin to execute **PWDPRINTBATCH** since there is no email ID provided to communicate the password:
  - The generated file is directly available in the path specified in property name **REPORT_SAVE_PATH** under **ApplicationWorkingDirectory/data/FBAReports.properties** file

- The generated password file is password protected, and the format to open the password file will be the value of **Lastname_Firstname** column in **CUSR** table

Difference Between TBA_DEBUG_LEVEL and TBA_DEBUG Parameters

Product: Finacle Core Banking (Connect24)

The **TBA_DEBUG_LEVEL** and **TBA_DEBUG** parameters are exported for debugging purposes in Finacle Core Banking. While both the parameters seem related, they have different functionalities. It is important to understand the difference between these parameters since they may seem inter-related and can appear to have the same functionality.

TBA_DEBUG_LEVEL is used to set the level of debugging logs generated in the **.evt** or **Event** file. The maximum level of debugging is 9, while the minimum value is 1. This parameter is usually deployed in the **Start-Uniser** or **Uniser** configuration file.

TBA_DEBUG is used for tracking the scenarios where the account gets locked during concurrent posting of transactions. The **ACID** of the account that is locked and the number of times it has been locked is recorded in the **PostLockAcct*** file in the **/cdci_logs/** directory. This parameter is also deployed in the **Start-Uniser** or **Uniser** configuration file.

For both the parameters, the configuration files in the **fce** directory can be decrypted, after which the changes can be made and encrypted. These changes made can take effect once the **UNISER** is restarted.

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