ORACLE

Data Sanity-Check Results Handbook

Oracle Financial Services Lending and Leasing

February 2023, Version 1.1

Public

Purpose statement

This document provides an overview of Data Sanity-Checks provided by Oracle Financial Services Lending and Leasing. It is intended solely to help with detailed information of Data Sanity-Checks for analysis and to plan your I.T. projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

Introduction

Data Sanity-Check Results or DSR are scripts with specific procedures. These scripts are executed by batch jobs to check for any mismatch in the customer data maintained in production to the kind/type of data that can be handled by Oracle Financial Services Lending and Leasing application.

Purpose

Sanity check is a key factor in the data-wrangling process when data discrepancy issues exist. Identifying, analyzing and resolving the issues can help for optimum functioning of the system and reduce downtime specifically during backend processing of data. Though sanity checks helps to identify accounts with mismatch in data, it does not report all production issues.

DSR Scheduling and Execution

DSR scripts provided by OFSLL can be used to pre-validate accounts data before migration and even after migration when new accounts are on-boarded.

DSR scripts are handled by specific batch jobs defined in OFSLL and can be can be executed by the Job Scheduler based on pre-set frequency and/or from User Interface. The following sections provide detailed information on the same.

Pre-requisites

Following are the pre-requisites while validating accounts data with DSR:

- The required Seed Data for DSR batch jobs are pre-configured by OFSLL during the installation and can be directly executed. Contact OFSLL support if SET-DSR jobs are not available in Batch Job Sets.
- Data Sanity check procedures are supported from OFSLL release 14.10.0.0.0 onwards only.
- By default, the frequency of DSR batch jobs are configured to run 'DAILY' and can be further configured by your system administrator based on requirement. However, it is recommended to run the DSR checks once in a month.
- By default, the DSR batch jobs are configured for ALL Company and Branch. Based on requirement, the same has to be customized for individual Company-Branch combinations.

Procedures in DSR Batch Jobs

The below table indicates the list of sanity check procedures with their purpose / description.

***Note**: The below list is iterative and will be enriched during subsequent updates.*

SL	PROCEDURE	PURPOSE / DESCRIPTION
1	VALIDATE_MISSING_GLTXNS	To check for accounts where the Transactions Codes were missing or NOT setup properly in GL
2	ACC_DUE_DT_NEXT_NULL	To check for accounts where the Next Due Date is MISSING/NULL
3	VALIDATE_PAYMENT	To check for accounts if there is repayment mismatch between number of payments and total term

SL	PROCEDURE	PURPOSE / DESCRIPTION
4	ACC_DUE_DLQ_VS_ADH_BALANCE	To check for accounts with mismatch between Total Due* and Due Balance in Due Date History. Here the *Total Due includes current due & excludes fees AND expenses.
5	ACC_RECEIVABLE_VS_ADV_BALANCE	To compare ACCOUNT RECEIVABLE BALANCE with ADVANCE BALANCE in account balances. This procedure is applicable for Loan Account Only.
6	VALIDATE_ACCOUNTS	This procedure performs the following functions: 1. To check SCRA for non-primary customers but SCRA accounts 2. To check if account rate run date next (ACC_RAT_RUN_DT_NEXT) is defined for matured accounts 3. To check if there are accounts with NO BILL IN THE PAST (LTC RUN DT - Late Charge Run Date) 4. To check if account delinquency date (ACC_DLQ_DT) is NOT in sync with delinquency condition open date 5. To check if the paid ahead but delinquency amount is greater than zero 6. To check if the ACC_DUE_DT_OLDEST date does not match with DUE DATE HISTORY for long past due accounts 7. To check if the account due delinquency amount (ACC_DUE_DLQ_AMT) is missing for delinquency accounts 8. To check if there are accounts with Credit Balance 9. To check if the account last extension date (ACC_EXTN_DT_LAST) is NOT NULL with NO extensions PAID AHEAD with Payment Date in Future
7	VALIDATE_BILLING	To check for accounts where billing did not happen on the first payment date AND/OR accounts for which billing was missed for a particular due date
8	VALIDATE_DELQ_ACCPAID	To check for accounts where DELINQUENT DAYS = 0 and having 5 Due Buckets with due amounts but the PAID AMOUNTS are populated as 0
9	VALIDATE_MATER_ACC_STMT	To check if the Billing Cycle OR Due Day is same between Master and Associated Account with Statement Consolidation Flag set to Y
10	PAID_ACCS_WITH_ACC_OUTSTANDING	To check for accounts which are paid but still has outstanding balance i.e. ACCOUNT RECEIVABLE NOT MATCHING WITH ADV BALANCES
11	VALIDATE_ACC_TXN	To check for accounts with mismatch in OLDEST DUE DATE
12	VALIDATE_PMTS_DUES_ALLOCATION	To check for accounts where the payment is received and balance is updated but no dues are allocated to dues buckets
13	VALIDATE_SCRA	This procedure performs the following functions: 1. To check for customers having both SCRA and NON-SCRA accounts 2. To check for SCRA customers who do not have a repayment schedule
14	VALIDATE_DELINQUENCY	To check if the account delinquency is not NULL for non-delinquent accounts
15	ACC_MATURITY_DT_CUR_NULL	To check for accounts where the Maturity Date is MISSING/NULL
16	VALIDATE_BALANCE	To check for accounts having negative balances in TRANSACTION TYPE CODE
17	VALIDATE_NON_MATURED_ACCOUNTS	To check for non-matured accounts where the DDT RUN DATE NEXT is set DT_MAX
18	VALIDATE_BANKRUPTCY	To check for accounts with multiple bankruptcy records for same customer

SL	PROCEDURE	PURPOSE / DESCRIPTION
19	VALIDATE_AMTDUE_OUTSTANDINGBAL	To check for accounts where Amount Due is greater than the Outstanding Balance
20	VALIDATE_DUE_DATES	To check for accounts with duplicate Due Dates
21	VALIDATE_BKRP_IND	To check for accounts where the CUS_BANKRUPTCY_IND = N even if bankruptcy record EXISTS for that customer
22	VALIDATE_TELECOM	To check for accounts which does not have active telecom details
23	VALIDATE_LTCRUNDATE_BACKDATE	To check for accounts where the LTC Run date Next is less than Back Dated Date
24	VALIDATE_POOL_PIPE	To check for valid securitization accounts in the pool for processing
25	VALIDATE_DUP_DUEDATE	To check for accounts having duplicate due dates
26	ACC_DUE_DAY_CUR_VS_MATDUE_CUR	To check for accounts where the MATURITY DUE DAY is not matching with CURRENT DUE DAY

Sequence and Frequency of DSR Batch Jobs

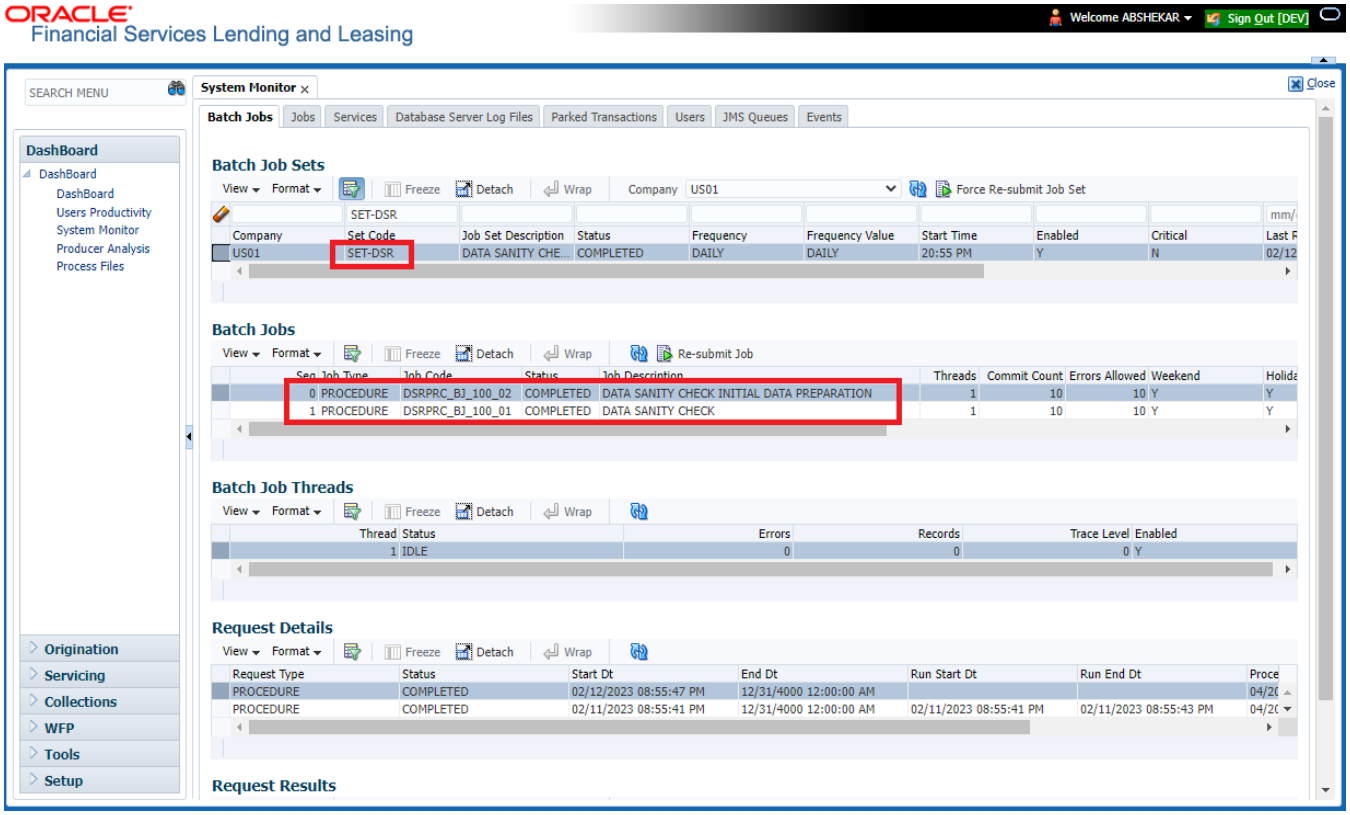
Sequence of DSR Batch Jobs

The sanity check procedures are available in SET-DSR Batch Job Set and are contained in two batch jobs - DSRPRC_BJ_100_01 and DSRPRC_BJ_100_02.

These batch jobs are to be executed in the following order:

First run the DSRPRC_BJ_100_02 under SET-DSR job set and only after successful completion, run the second batch job DSRPRC_BJ_100_01.

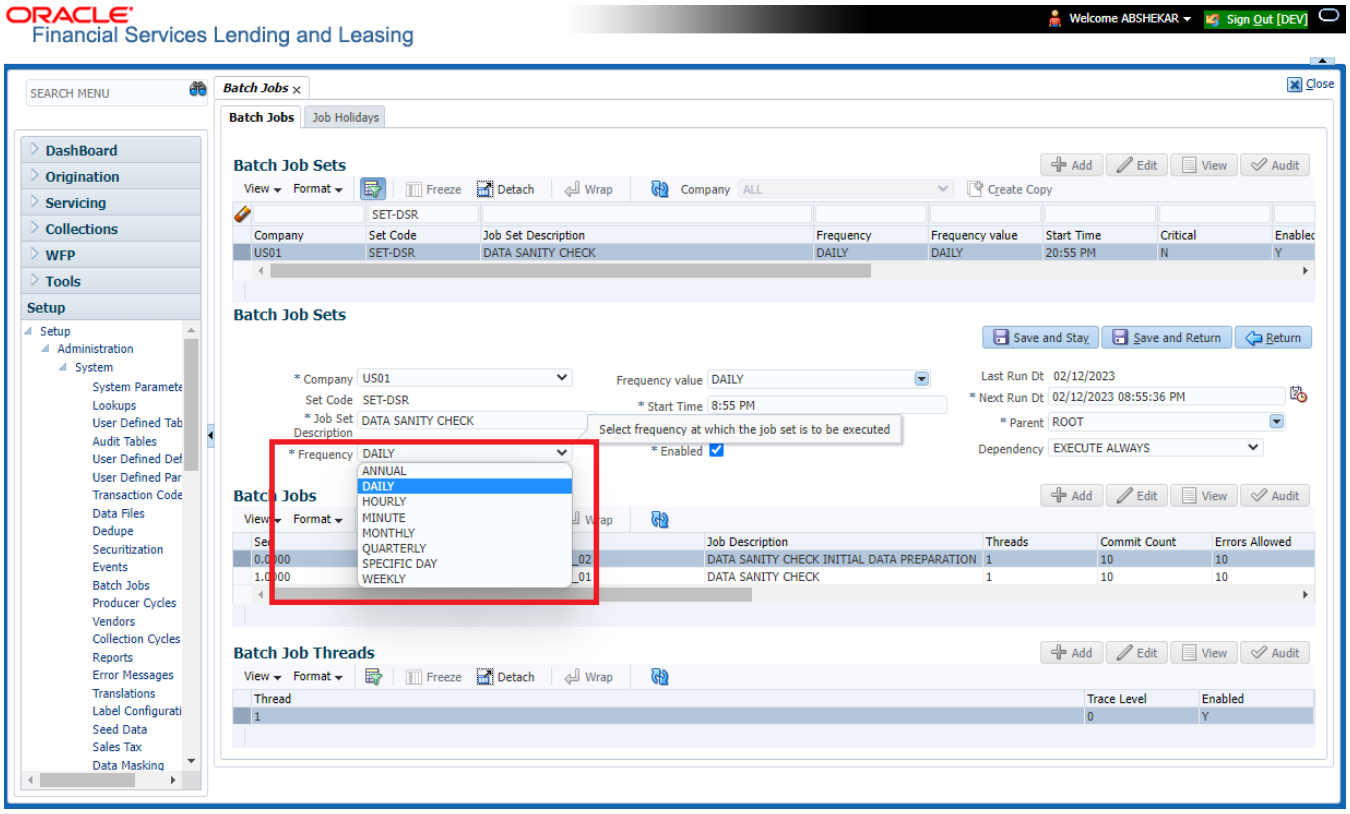
The order is pre-defined as 0 and 1 in the sequence as indicated below. In OFSLL application, navigate to Dashboard > System Monitor > Batch Jobs tab to view the DSR batch jobs.



Frequency of DSR Batch Jobs

The frequency of the DSR batch jobs are to be configured based on requirement. However, by default in day zero setup, the frequency is set to 'DAILY'.

1. In OFSLL application, navigate to Setup > Administration > System > Batch Jobs screen.
2. In Batch Job Sets section, filter and select the SET-DSR batch job.
3. Click Edit and set the Frequency as indicated below. Save the record after update.



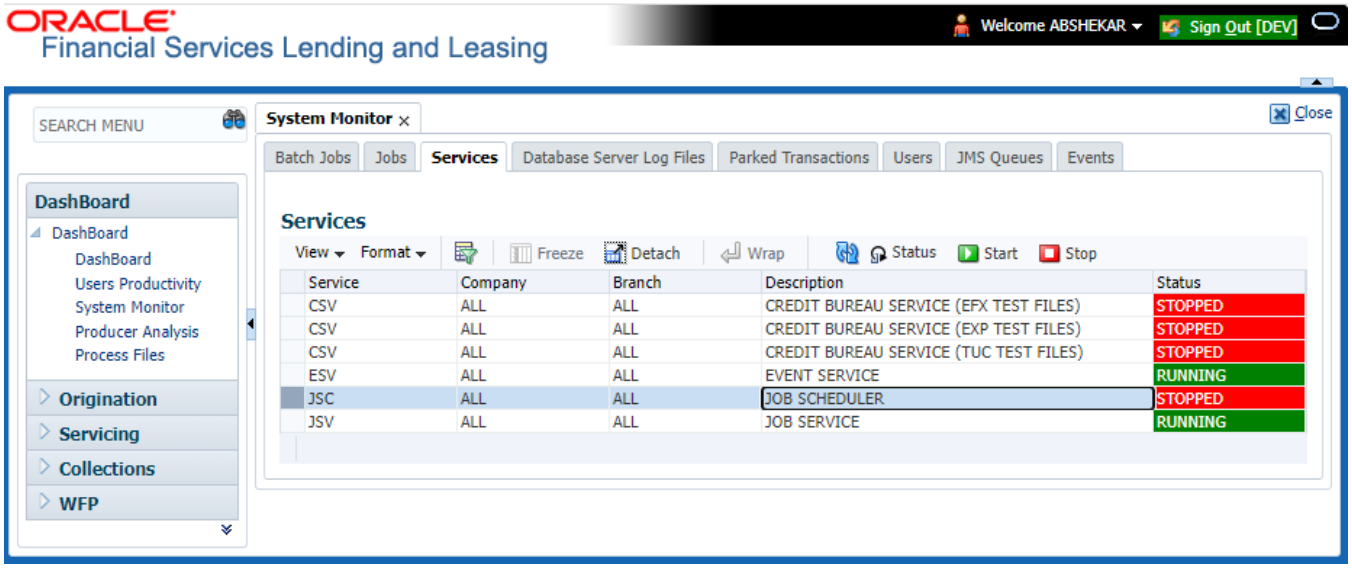
How to Execute DSR Batch Jobs

DSR batch jobs can be executed either from Job Scheduler or by Application Interface.

Via Job Scheduler

The job scheduler automatically processes the scheduled jobs based on 'Next Run Date' defined in Batch Job Set definition. The Services screen allows you to track and maintain all the system's processing services including batch job scheduler. System administrator can start or stop a service on this screen using the action buttons respectively.

To stop, start or refresh a processing service, click Dashboard > Dashboard >System Monitor > Services.



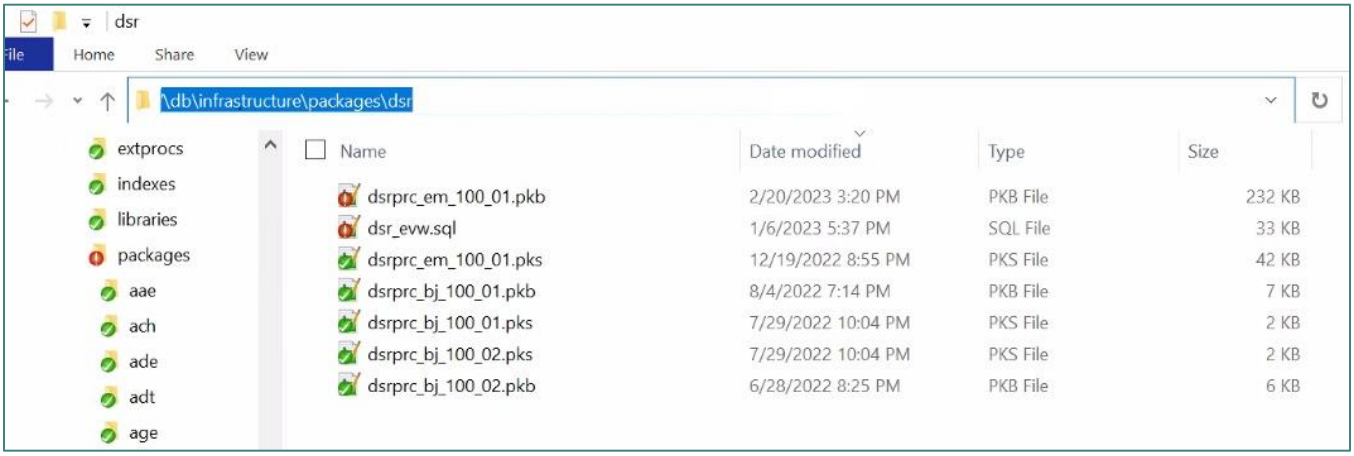
Via Application Interface

1. In OFSLL application, navigate to Dashboard > System Monitor > Batch Jobs tab.
2. Select the required batch job based on the order defied in 'Seq' column. For more details, refer to Sequence of DSR Batch Jobs section.
3. Click **Re-Submit Job**.
4. Once done, verify the execution details in Request Results section.

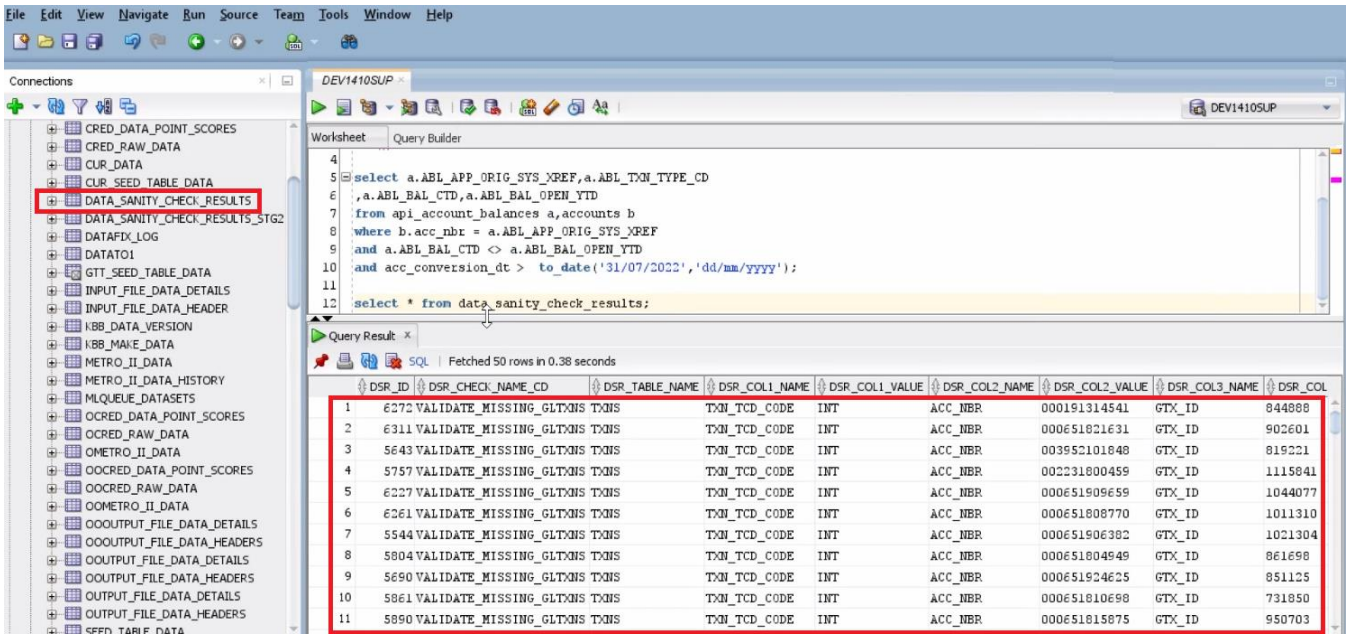
View Data Sanity-Check Results

After the Data Sanity-Check validation is complete, the results are stored in the following location:

<<OFSLL HOME>>/db/infrastructure/packages/dsr



All the validated data can be accessed by querying the DATA_SANITY_CHECK_RESULTS table or dsr_dsr_evw view for further analysis.



Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle