Customized Transaction to Trigger Process Chain from Failed Step

Applies to:

SAP BW 3.x & SAP BI NetWeaver 2004s. For more information, visit the Business Intelligence homepage.

Summary

There are multiple process chains running in the production system at a given time. During the execution of the process chain, some of the process variant may fail due to various reasons. We require correcting these process variant and triggering the chain ahead from next step so that the execution of chain is completed.

The document explain the step to manually trigger the chain from failed process type and steps in creating customized transaction to perform the task via a single TCODE

Author: Shakir Iqbal Kapdi

Company: Infosys Technologies LTD

Created on: 12 April 2010

Author Bio

Shakir Kapdi has more than 3 years of experince in SAP. He has worked on various project like implementation, Upgarde, Application support and Enhancment. He has experince in SAP BI 3.X and 7.0. He is currently working in Infosys Technologies LTD.

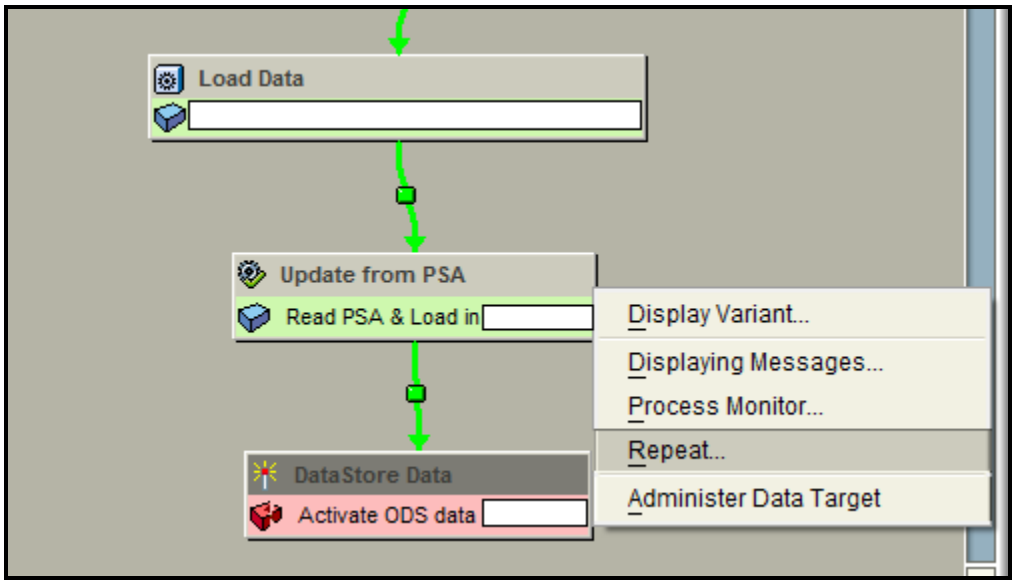
Table of Contents

Steps to Trigger the Chain from Failed Step 3
Steps to create customized transaction 9
Using the Customized TCODE 14
Disclaimer and Liability Notice 15

Steps to Trigger the Chain from Failed Step

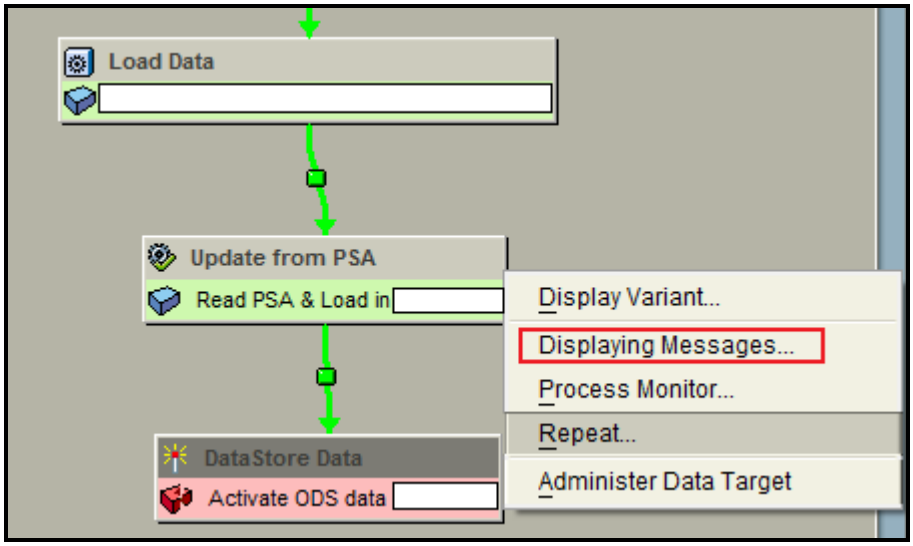
During the execution of the process chain, there might be failures in some of the process type. The simple way of correcting the error would be repeating the step in the process chain, this would not only repeat the step but also trigger the chain ahead on successful completion of the step.

To repeat a step in process chain, *right click* on the failed process and click ***"Repeat" or "Repair"*** as shown below.

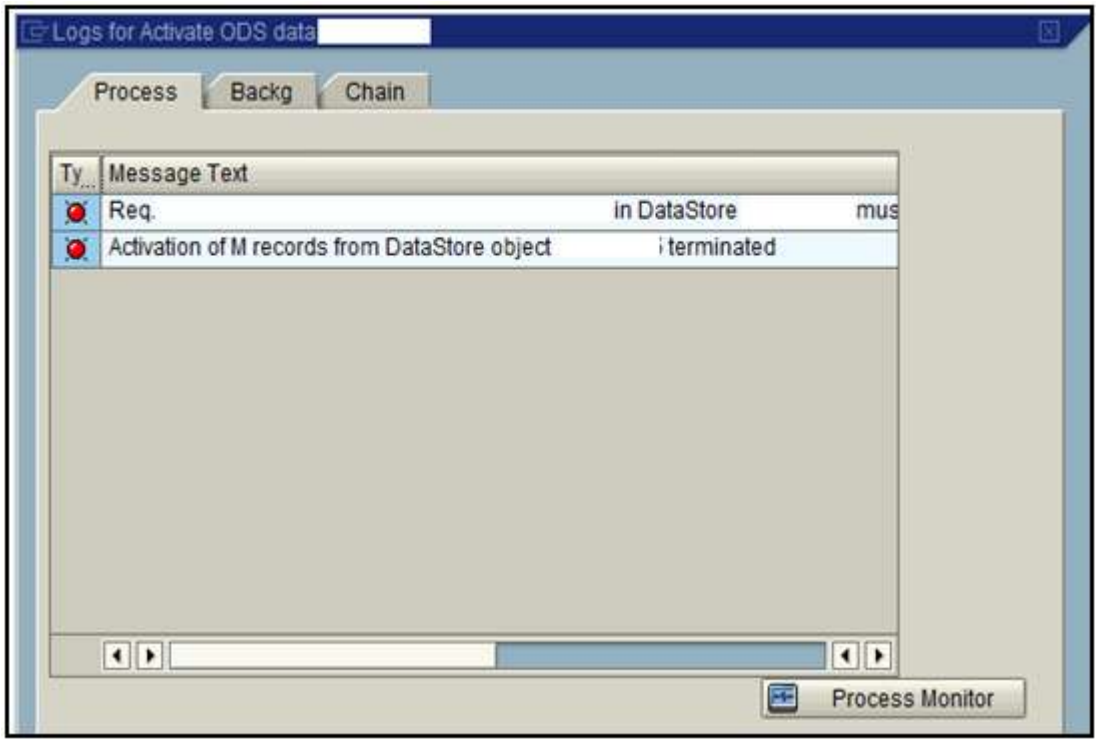


However, there would be scenario where the task associated with the process type is carried out manually and we just require triggering the process chain ahead from next step rather than doing the repeat. For example activation of request in DSO or rollup of request in the cube which can be done from the manage tab of the objects.
Following step requires to be followed to trigger the process chain ahead of the failed step manually.

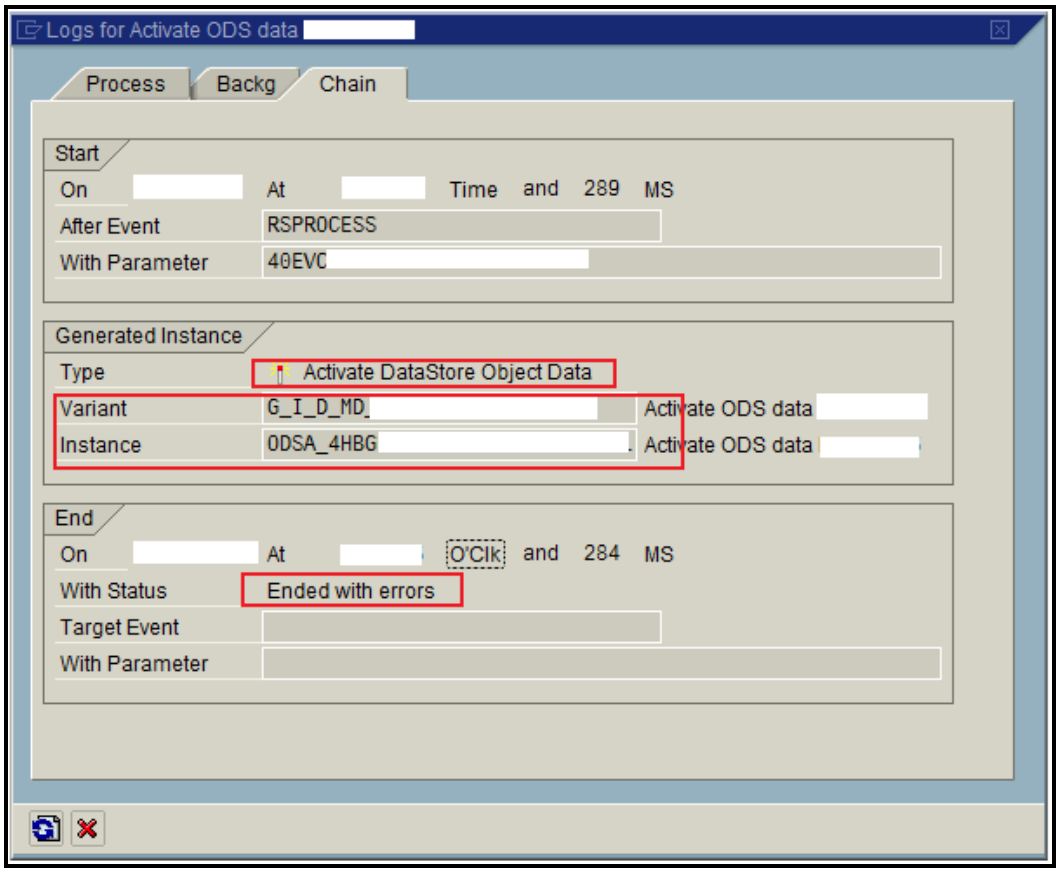
1) Right click on the failed process type and then click on the ***"Display Messages"*** tab.



This will give you the details of the process.

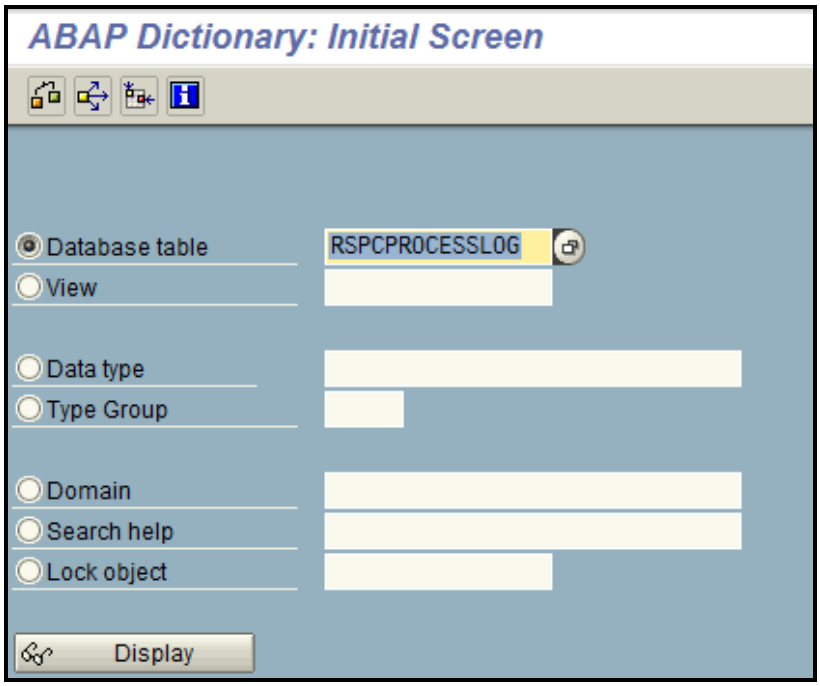


2) Then click on the ***Chain*** tab. This will provide you the detail of the variant in that particular run.

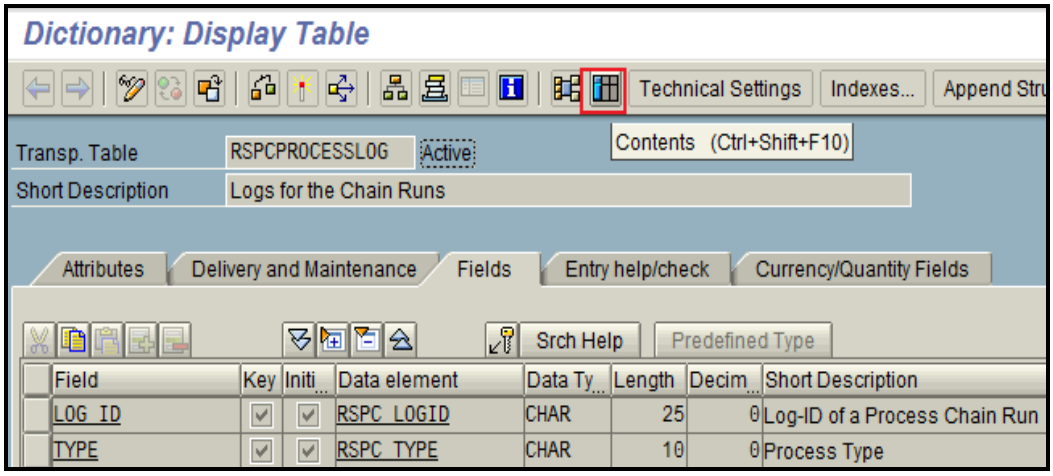


Note down the Variant and the Instance details.

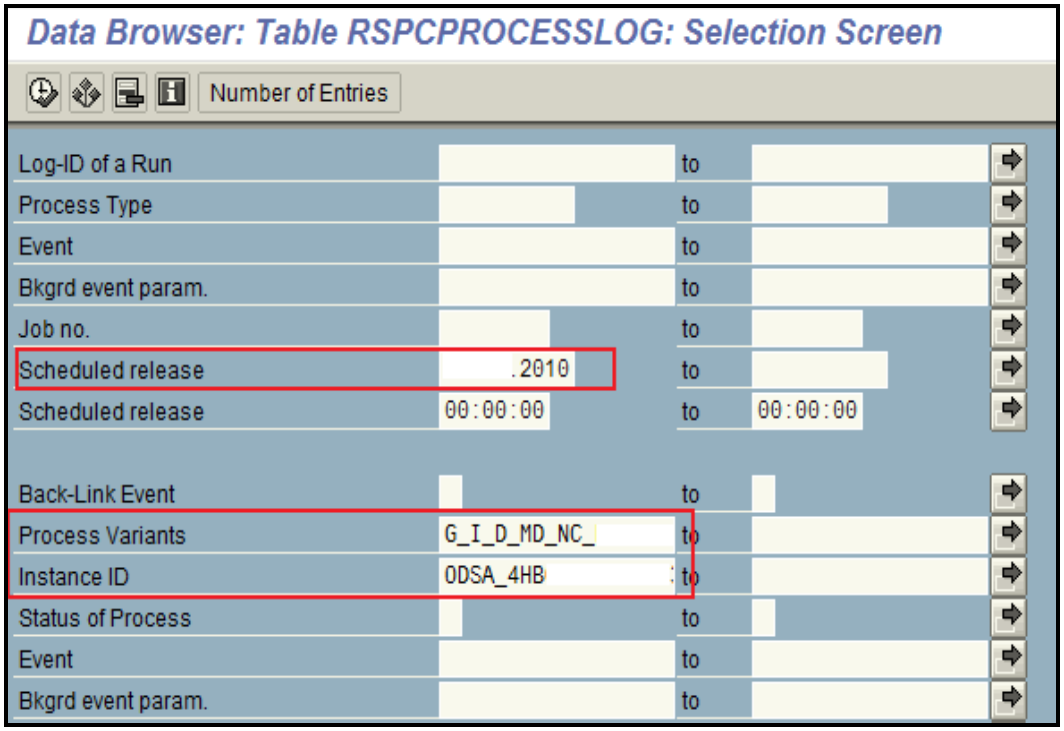
3) In another session, open the transaction ***SE12***. In the data base table field, enter the table name ***"RSPCPROCESSLOG"*** and then click display. This table contains the details (logs) of the chain run.



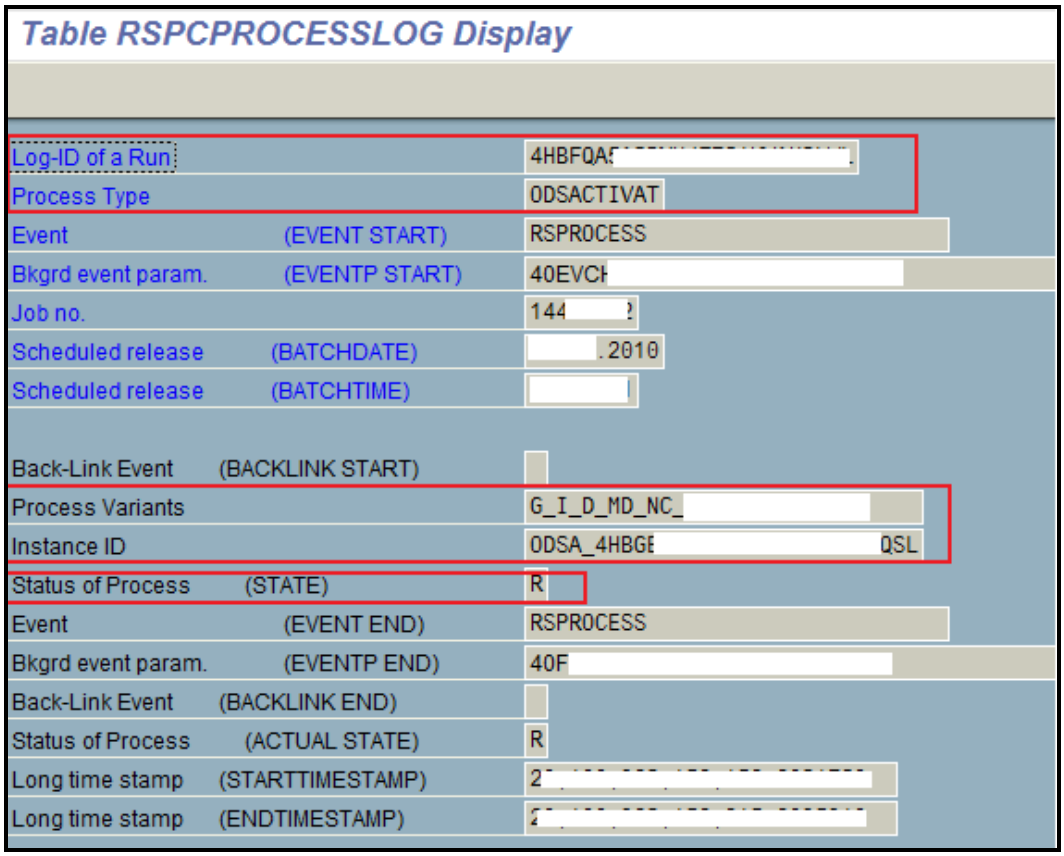
4) Click on the "content" button as shown below to see the content of the table.



5) In the selection option, provide the details as shown below.



The process variant and the Instance are same as that explained in step 2. The scheduled release date is same date as of execution of the chain. On executing, we will get further logs of the process variants as shown below

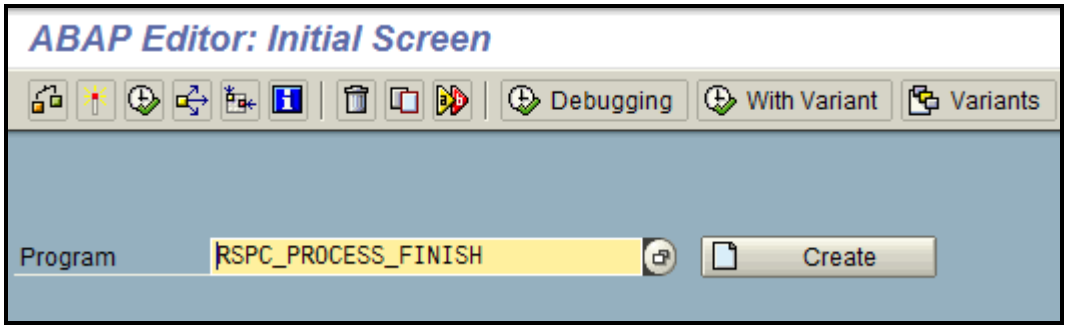


Out of the above details, the ***"Log ID of Run"*** the ***"process type", "process variant" and "Instance ID"*** are required for further processing.

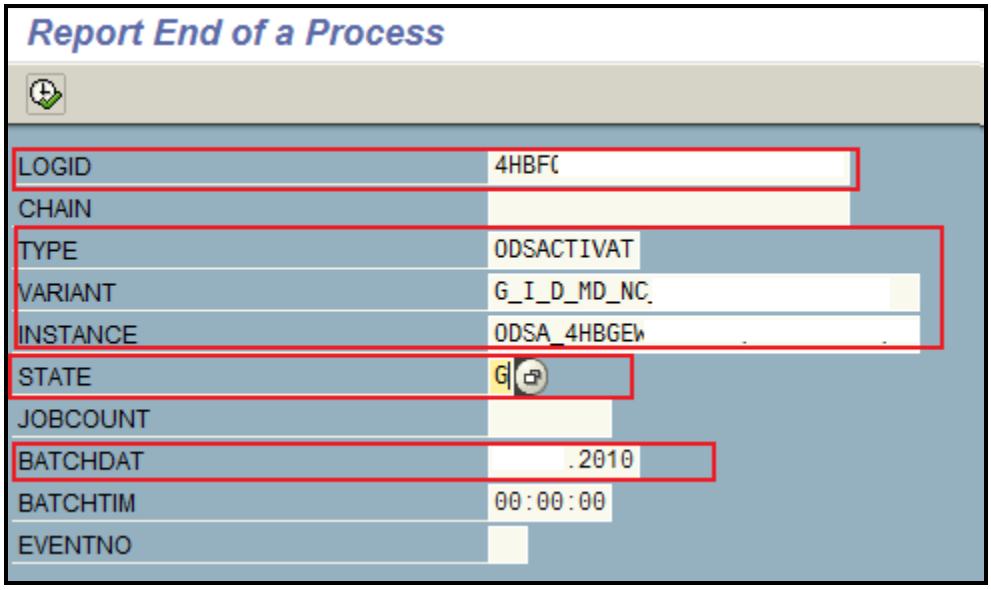
The process variant and instance ID are same as in step 2.

We can also see that the status of process chain is ***"R"*** which means ended with error.

6) In another session, go to transaction ***"SE38"***. In the program field enter the program name ***"RSPC_PROCESS_FINISH"*** this is the program used to trigger the process chain. It changes the status of the process variant from "R" to "G" means completed, thus triggering the chain.



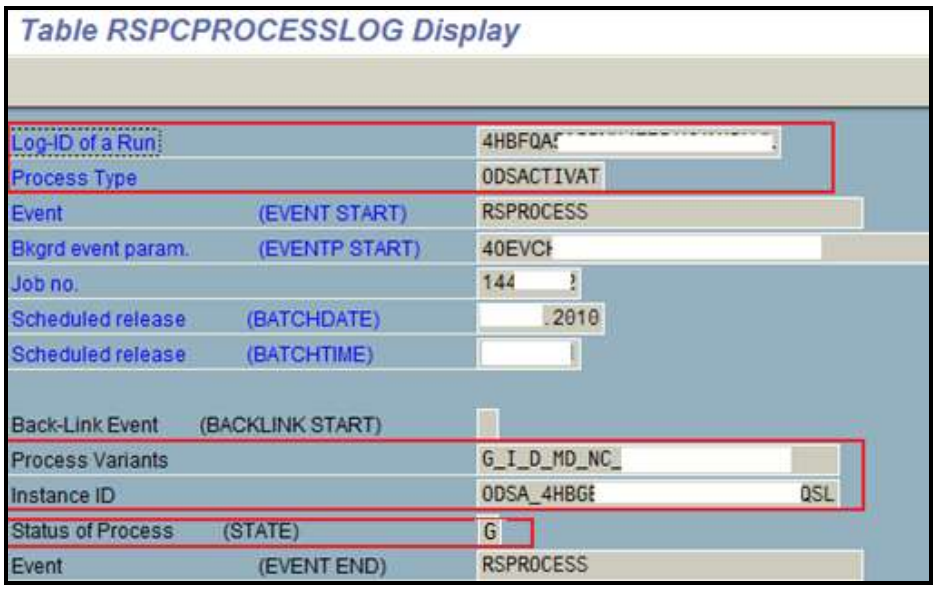
7) In the selection screen of the program enter the details as shown below and execute.



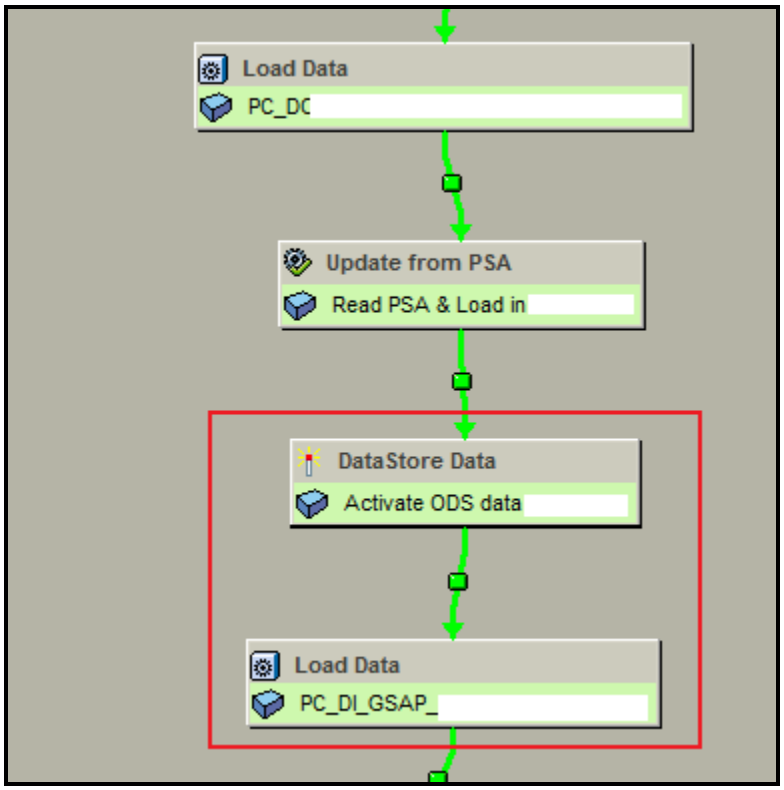
"LOGID", "TYPE", "VARIANT", "INSTANCE", "BATCHDAT" are same as copied in step 5.

The ***"STATUS"*** *field is set to* ***"G".***

8) This will change the status of the chain and will trigger the chain ahead. The status is also reflected in the table ***"RSPCPROCESSLOG".***



In addition, the process chain is triggered ahead.

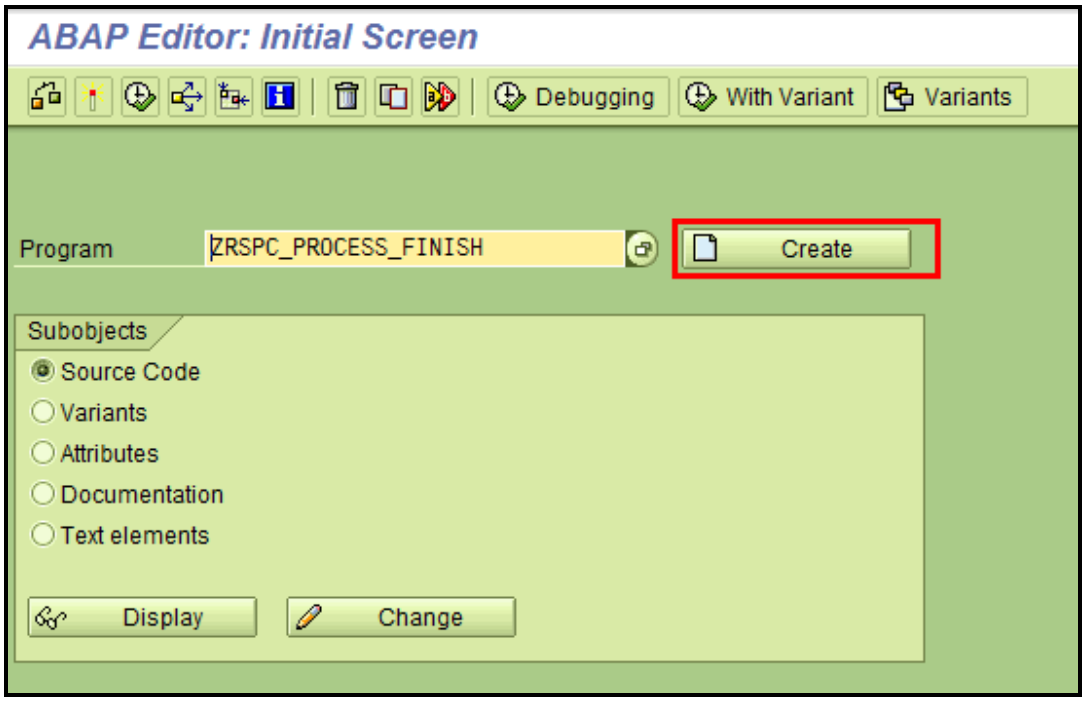


These are the step, which are required to be followed in case we need to trigger the failed process chain ahead manually.

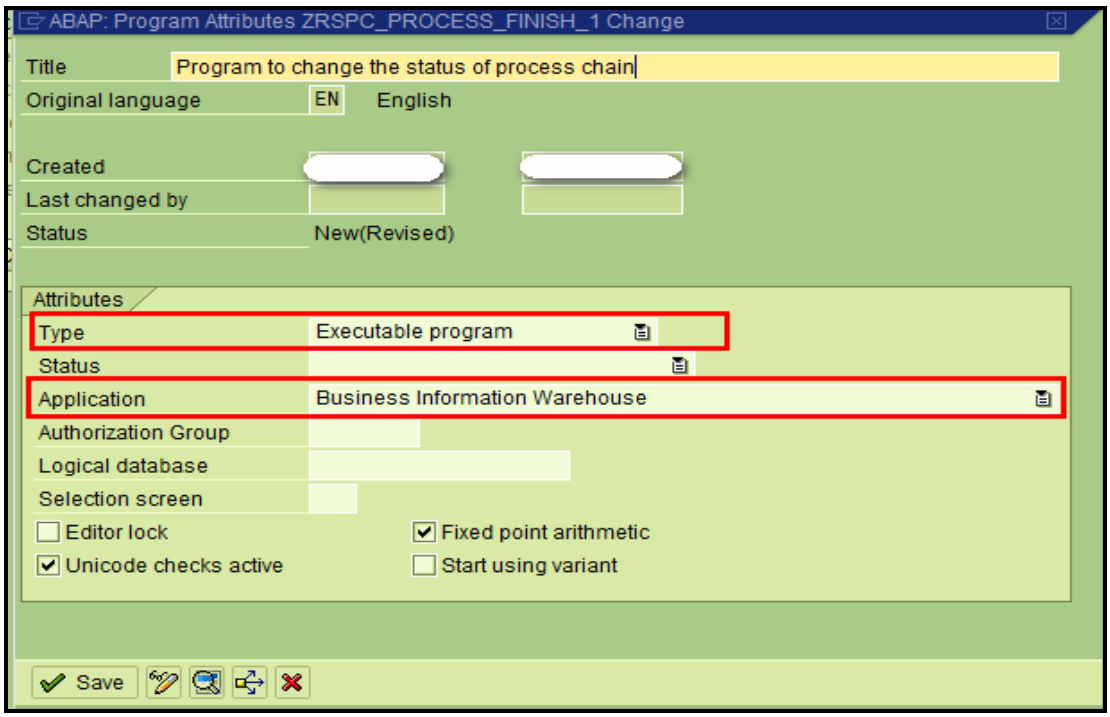
The above manual step requires navigation from multiple sessions and noting different details. We can create a customized transaction to achieve this functionality

Steps to create customized transaction

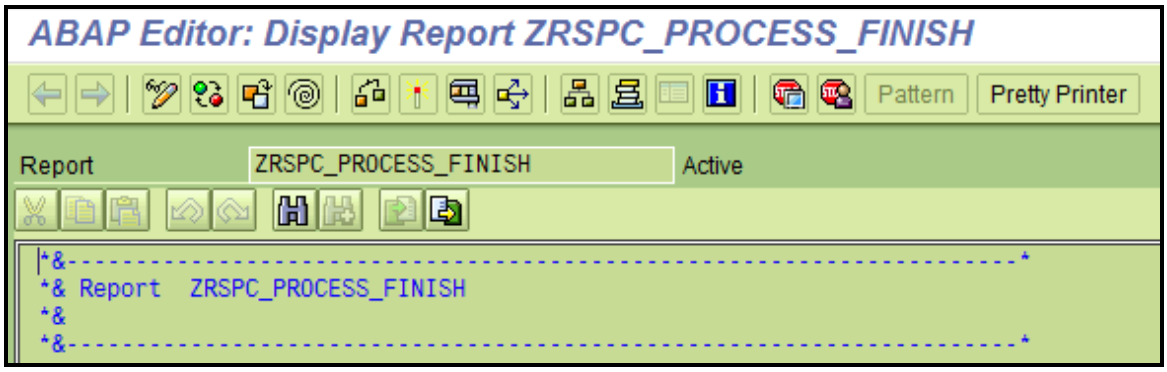
1. Go to transaction SE38. Give a program name and click on create button. In our case, the program name is “ZRSPC_PROCESS_FINISH”.



2. Provide a title to the program. Select type as “Executable”. Select application as “Business information warehouse” and then save.



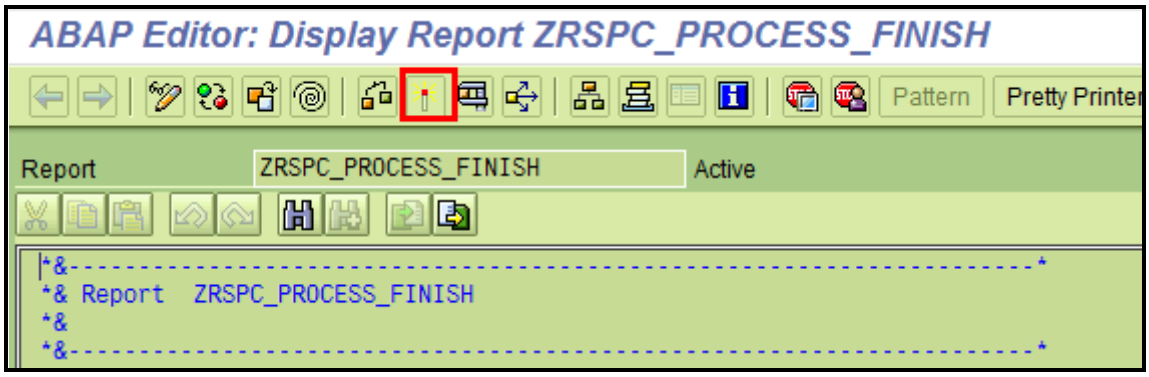
3. In the ABAP editor as shown below copy the below code



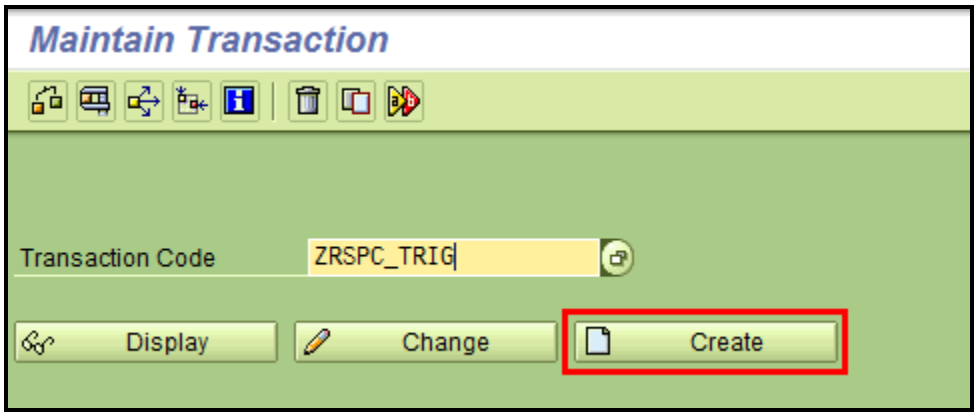

```
***************************** -Code- *********************************
*&---------------------------------------------------------------------*
*& Report  ZRSPC_PROCESS_FINISH
*&
*&---------------------------------------------------------------------*

REPORT  ZRSPC_PROCESS_FINISH.

*&---------------------------------------------------------------------*
*& declaration of variables/Data type
*&
*&---------------------------------------------------------------------*

DATA: VAR  TYPE rspc_variant,
INS  TYPE rspc_instance ,
DAT  TYPE SY-DATUM,
STAT TYPE rspc_state.

DATA: I_PCLOG like RSPCPROCESSLOG.

*&---------------------------------------------------------------------*
*& declaration of parameters for selection screen
*&
*&---------------------------------------------------------------------*

PARAMETERS:VARIANT  TYPE RSPC_VARIANT OBLIGATORY,            "Stores variant of PC
INSTANCE TYPE RSPC_INSTANCE OBLIGATORY,           "Stores instance of PC
DATE     TYPE SY-DATUM OBLIGATORY,                "Stores batch date of PC
STATE    TYPE RSPC_STATE OBLIGATORY DEFAULT 'G'.  "Status by Default 'G'


*&---------------------------------------------------------------------*
*& Logic to trigger the chain
*&
*&---------------------------------------------------------------------*

VAR = VARIANT.
INS = INSTANCE.
DAT = DATE.
STAT = STATE.
```

```
SELECT single * FROM RSPCPROCESSLOG
INTO  I_PCLOG
WHERE VARIANTE = VAR
AND INSTANCE = INS
AND BATCHDATE = DAT.

IF SY-SUBRC = 0.


CALL FUNCTION 'RSPC_PROCESS_FINISH'
EXPORTING
I_LOGID                = I_PCLOG-LOG_ID
*     I_CHAIN                  =
I_TYPE                 = I_PCLOG-TYPE
I_VARIANT              = I_PCLOG-VARIANTE
I_INSTANCE             = I_PCLOG-INSTANCE
I_STATE                = STATE
*     I_EVENTNO                =
*     I_HOLD                   =
*     I_JOB_COUNT              =
I_BATCHDATE            = I_PCLOG-BATCHDATE.
*     I_BATCHTIME              =
*     I_DUMP_AT_ERROR          =


ELSE.

MESSAGE 'Process selected does not exist - Check your entry' TYPE 'I'.

ENDIF.
```

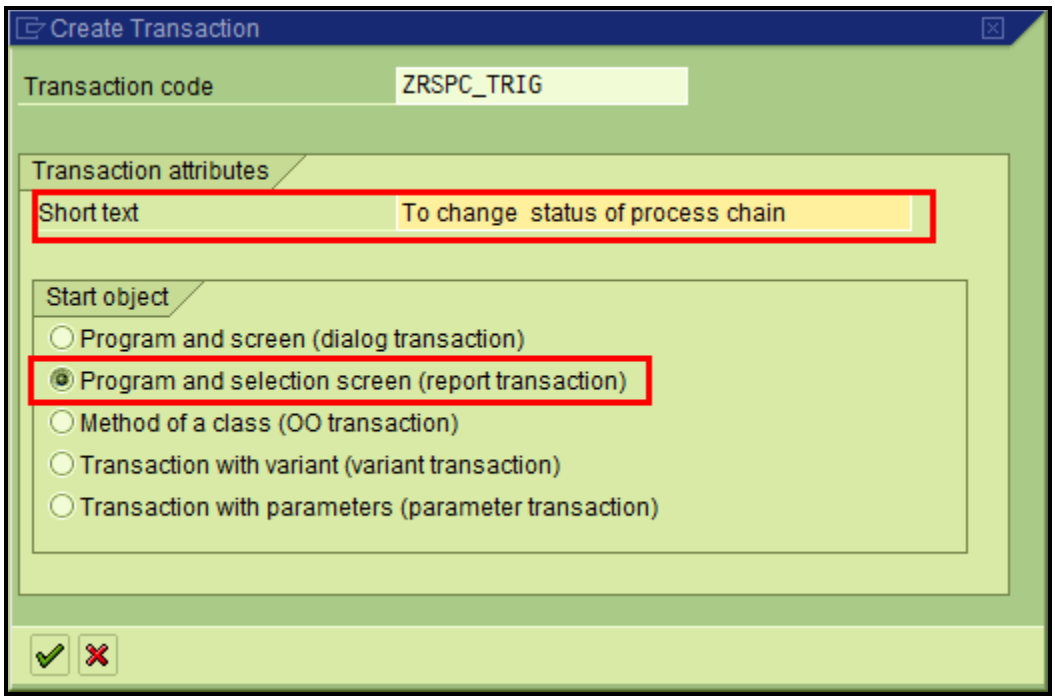
4. Activate the program.



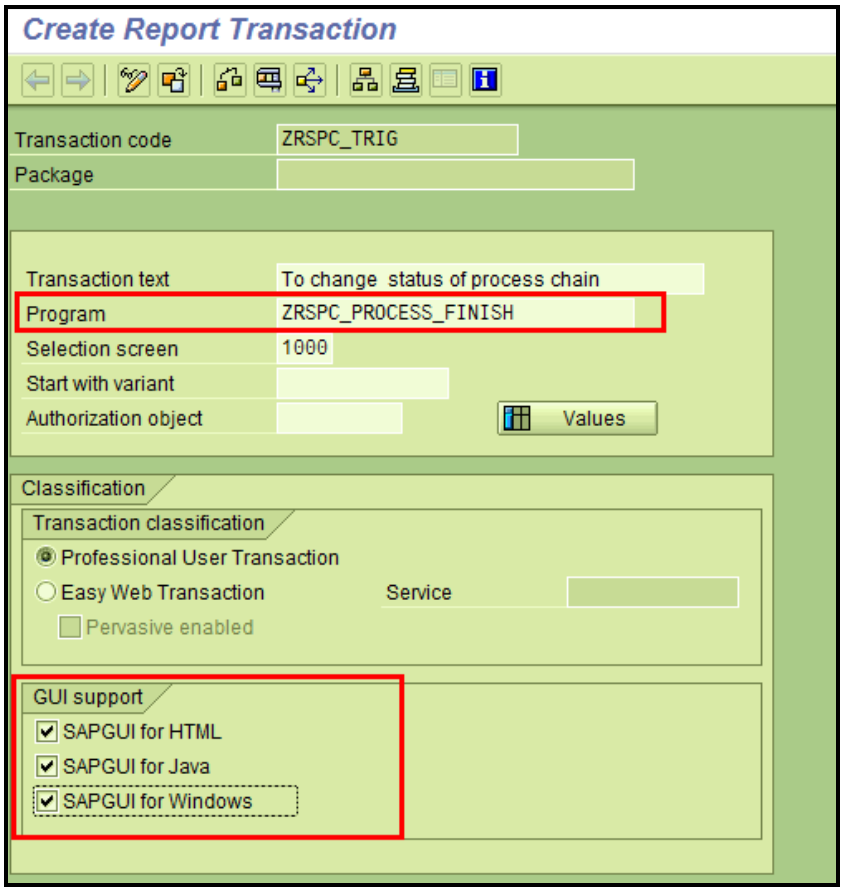
5. Go to transaction SE93 and create a TCODE, in our case ZRSPC_TRIG.



6. Provide Short text (description) for the TCODE and select the type as “Program and selection screen (report transaction)”.



7. Provide the technical name of the program that requires being associated with the transaction. The program name would be same as that created in step1 i.e. “ZRSPC_PROCESS_FINISH”. Select all the GUI support.



8. Save the TCODE.

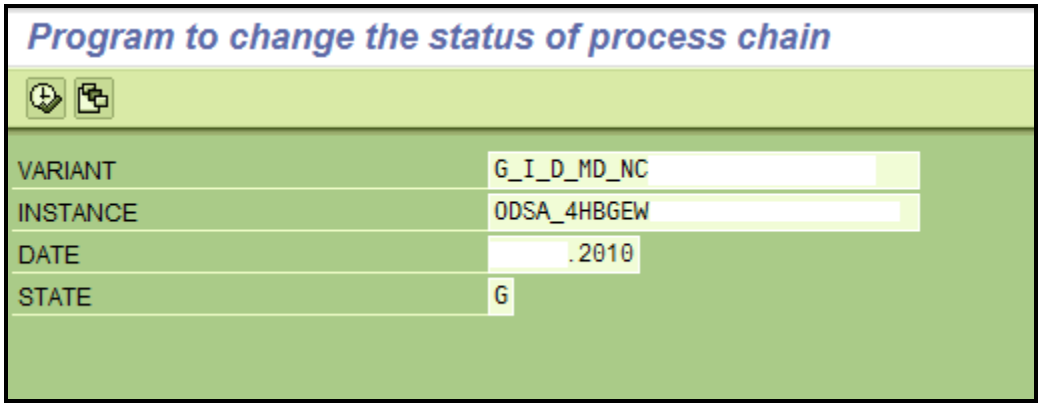
These complete the step to create a customized transaction code.

Using the Customized TCODE

Consider the scenario as explained in the manual triggering procedure. The process chain has failed at activation step, which has been corrected manually. We now require triggering the process chain ahead from the failed step.

We will follow the first two-step as explained in the manual process i.e. noting down the variant and the instance.

Execute the TCODE ZRSPC_TRIG and provide the details.



Execute the program to trigger the chain.

The creation of customized TCODE reduce the effort of looking up the data to different table and navigation to different session.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.