

# Maintaining Data Quality in BW using Error Stack



## Applies to:

SAP BW 7.0. For more information, visit the [Business Intelligence homepage](#).

## Summary

System maintains the data quality by imposing checks for incorrect data records at various places while loading data into BW. Apart from the various Checks imposed by system for Incorrect Data records we can also mark a data record as incorrect/invalid and can move the record to error stack by writing ABAP code in transformations (start, end, characteristic or expert routines), By doing so we can maintain the quality of data that is being loaded to BW Data Targets. This Article explains the different ways in which the incorrect data records can be moved to error stack when the data record is processed in the routines(Start, End, Characteristic or Expert Routines) of Transformation (In this case a data record is marked as incorrect based on Customer-specific requirements or Conditions).

**Author:** Ramakrishna Gattikoppula

**Company:** Infosys Technologies Ltd.

**Created on:** 10 August 2010

## Author Bio

Ramakrishna Gattikoppula is a Senior Systems Engineer at Infosys Technologies Ltd. He has been involved in SAP BW implementation projects.

## Table of Contents

Introduction .....	3
Business Scenario: .....	4
Moving Records to Error Stack:.....	4
1. In Characteristic Routine:.....	4
2. in Start/End Routine:.....	5
3. In Expert Routine: .....	5
Related Content .....	6
Appendix:.....	7
Characteristic Routine Code: .....	7
Start Routine Code: .....	7
End Routine Code: .....	8
Disclaimer and Liability Notice.....	9

## Introduction

System maintains the data quality by imposing checks for incorrect data records at various places while loading data into BW.

Following is the table which provides the overview of where checks for incorrect data records.

Where Is Check Run?	Examples of Incorrect Data Records
In the transformation	<ul style="list-style-type: none"> <li>• Field contains invalid characters or lowercase characters</li> <li>• Error during conversion</li> <li>• Error during currency translation</li> <li>• A routine returns a return code &lt;&gt; 0</li> <li>• Characteristic value is not found for master data</li> <li>• Error reading master data</li> <li>• Customer-specific formula results in error</li> </ul>
When data is updated to the master data table or text table	<ul style="list-style-type: none"> <li>• Invalid characters in keys or navigation attributes</li> <li>• If no SID exists for the value of the navigation attribute</li> <li>• If no language field is defined for texts</li> <li>• Invalid "from" and "to" dates</li> <li>• Duplicate data records with relation to keys</li> <li>• Overlapping and invalid time intervals</li> </ul>
When data is updated to the InfoCube	<ul style="list-style-type: none"> <li>• If no SID exists for the characteristic value</li> </ul>
When checking referential integrity of an InfoObject against master data tables or DataStore objects	<ul style="list-style-type: none"> <li>• If no SID exists for the characteristic value</li> </ul>

Apart from the various Checks imposed by system for Incorrect Data records we can also mark a data record as incorrect/invalid and can move the record to error stack by writing ABAP code in transformations (start, end, characteristic or expert routines), By doing so we can maintain the quality of data that is being loaded to BW Data Targets.

This Article explains the different ways in which the incorrect data records can be moved to error stack when the data record is processed in the routines(Start, End, Characteristic or Expert Routines) of Transformation (In this case a data record is marked as incorrect based on Customer-specific requirements or Conditions).

## Business Scenario:

Let us take an Example of a Training Report where you want to analyze the pass percentage of the students who have completed their training over a certain period of time along with their average score i.e. Parameters for this report would be Training Completion Date.

This report will give false analysis if we are getting the data with Completion Date as blank, so in order to have correct data for reporting we need to send the records back to get them corrected. This can be achieved by moving the erroneous records into Error Stack where they can be corrected and loaded to the Data Targets.

Above mentioned process of moving the data record into Error Stack can be done while processing the data in Transformations Routines.

Following are the Sample Codes for different types of routines (Start, Characteristic, End and Expert Routines) for the same example to push the records to Error Stack based on the condition on Completion Date:

## Moving Records to Error Stack:

### 1. In Characteristic Routine:

In Characteristic Routine we can raise an exception **CX\_RSROUT\_SKIP\_RECORD** to move the record into Error Stack. Along with raising exception we need to maintain a monitor entry so that it is easy to track the error record.

Following are the steps to push the error record into Error Stack in Characteristic Routine.

1. Set a monitor entry with error code (i.e. Message Type = E).
2. Raise an exception **CX\_RSROUT\_SKIP\_RECORD** to skip the entry.

Following is the Code for the above steps:

```

DATA:
  MONITOR_REC      TYPE rsmonitor.

*$$$ begin of routine - insert your code only below this line      *-*

* Direct Mapping if the Training End Date is not blank,
* Add to error stack if this field is blank.
  if SOURCE_FIELDS-trainendda is NOT INITIAL or SOURCE_FIELDS-trainendda <> ''.
    RESULT = SOURCE_FIELDS-trainendda.
  ELSE.
    monitor_rec-msgid = 'ZMESSAGE'.
    monitor_rec-msgty = 'E'.
    monitor_rec-msgno = '001'.
    monitor_rec-msgv1 = 'ERROR, Training End Date is Blank'.
    monitor_rec-msgv2 = SOURCE_FIELDS-trainendda.
    append monitor_rec to monitor.
    RAISE EXCEPTION TYPE CX_RSROUT_SKIP_RECORD.
  endif.

*$$$ end of routine - insert your code only before this line      *-*

```

With the above Characteristic Routine in the transformations you can find all those records in the Error Stack where the Training End Date is Blank.

## 2. in Start/End Routine:

Like in Characteristic Routine it is not possible to invoke CX\_RSROUT\_SKIP\_RECORD from start/end routine because this exception is not defined in the method for Start or End Routine.

Following are the steps to push the records into Error Stack while processing the data in start/End routine.

1. Set monitor entry with error code (i.e Message Type = E).
2. Set monitor entry with SKIPPED = X
3. Delete the Source/Result Package record.
4. Continue the Loop.

Following is the Code for the above steps:

```

DATA:
  MONITOR_REC      TYPE rsmonitor.

*$$$ begin of routine - insert your code only below this line      *--*

LOOP at SOURCE_PACKAGE INTO WA_SOURCE_PACKAGE.

* Direct Mapping if the Training End Date is not blank,
* Add to error stack if this field is blank.
  IF WA_SOURCE_PACKAGE-TRAINENDDA is INITIAL or
    WA_SOURCE_PACKAGE-TRAINENDDA = ''.
    monitor_rec-msgid = 'ZMESSAGE'.
    monitor_rec-msgty = 'E'.
    monitor_rec-msgno = '001'.
    monitor_rec-msgv1 = 'ERROR: Training End Date is Blank'.
    monitor_rec-msgv2 = WA_SOURCE_PACKAGE-TRAINENDDA.
    MONITOR_REC-RECNO = WA_SOURCE_PACKAGE-RECORD.
    MONITOR_REC-SKIPPED = 'X'.
    append monitor_rec to MONITOR.
    DELETE SOURCE_PACKAGE.
    CONTINUE.
  ENDIF.

ENDLOOP.

```

Code for the End Routine is same and is present in the Appendix, only difference is that instead of Source Package we will loop at Result Package.

## 3. In Expert Routine:

Following SAP Note has the procedure to move the records to error stack while processing the data using Expert Routine.

[Note 1227667 - Guidelines for expert routine: Design rules](#)

## Related Content

[http://help.sap.com/saphelp\\_nw70/helpdata/en/43/bcdb7801344defe1000000a422035/frameset.htm](http://help.sap.com/saphelp_nw70/helpdata/en/43/bcdb7801344defe1000000a422035/frameset.htm)

<http://forums.sdn.sap.com/index.jspa#4>

[Note 1349820 - Expert routine: Technical enhancements](#)

## Appendix:

### Characteristic Routine Code:

```

* Direct Mapping if the Training End Date is not blank,
* Add to error stack if this field is blank.
  if SOURCE_FIELDS-trainendda is NOT INITIAL or
  SOURCE_FIELDS-trainendda <> ''.
    RESULT = SOURCE_FIELDS-trainendda.
  ELSE.
    monitor_rec-msgid = 'ZMESSAGE'.
    monitor_rec-msgty = 'E'.
    monitor_rec-msgno = '001'.
    monitor_rec-msgv1 = 'ERROR, Training End Date is Blank'.
    monitor_rec-msgv2 = SOURCE_FIELDS-trainendda.
    append monitor_rec to monitor.
    RAISE EXCEPTION TYPE CX_RSROUT_SKIP_RECORD.
  endif.

```

### Start Routine Code:

```

LOOP at SOURCE_PACKAGE INTO WA_SOURCE_PACKAGE.

* Direct Mapping if the Training End Date is not blank,
* Add to error stack if this field is blank.
  IF WA_SOURCE_PACKAGE-TRAINENDDA is INITIAL or
  WA_SOURCE_PACKAGE-TRAINENDDA = ''.
    monitor_rec-msgid = 'ZMESSAGE'.
    monitor_rec-msgty = 'E'.
    monitor_rec-msgno = '001'.
    monitor_rec-msgv1 = 'ERROR: Training End Date is Blank'.
    monitor_rec-msgv2 = WA_SOURCE_PACKAGE-TRAINENDDA.
    MONITOR_REC-RECNO = WA_SOURCE_PACKAGE-RECORD.
    MONITOR_REC-SKIPPED = 'X'.
    append monitor_rec to MONITOR.
    DELETE SOURCE_PACKAGE.
    CONTINUE.
  ENDIF.

ENDLOOP.

```

**End Routine Code:**

```
LOOP at RESULT_PACKAGE INTO WA_RESULT_PACKAGE.

* Direct Mapping if the Training End Date is not blank,
* Add to error stack if this field is blank.
  IF WA_RESULT_PACKAGE-TRAINENDDA is INITIAL or
    WA_RESULT_PACKAGE-TRAINENDDA = ''.
    monitor_rec-msgid = 'ZMESSAGE'.
    monitor_rec-msgty = 'E'.
    monitor_rec-msgno = '001'.
    monitor_rec-msgv1 = 'ERROR: Training End Date is Blank'.
    monitor_rec-msgv2 = WA_RESULT-TRAINENDDA.
    MONITOR_REC-RECNO = WA_RESULT_PACKAGE-RECORD.
    MONITOR_REC-SKIPPED = 'X'.
    append monitor_rec to MONITOR.
    DELETE RESULT_PACKAGE.
    CONTINUE.
  ENDIF.

ENDLOOP.
```

## 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.