

# Guidelines for Effective Data Migration



## Applies to:

SAP R/3. All releases. For more information, visit the [ABAP homepage](#).

## Summary

Data Migration is an important step in any SAP implementation projects. This article will give you guidelines to be followed for effective data migration.

**Author:** Ankur Parab

**Company:** Capgemini Consulting India Pvt. Ltd.

**Created on:** 17<sup>th</sup> September 2009

## Author Bio

Ankur Parab has over 5 years of experience in SAP ABAP and an overall experience of 6 years in IT.

## Table of Contents

Purpose .....	3
Techniques.....	3
Procedure.....	3
Identifications .....	3
Decision .....	4
Development.....	5
Testing .....	8
Final Migration .....	11
Advantages / Disadvantages .....	11
Advantages of using LSMW:-.....	11
Disadvantage of using LSMW:-.....	11
Advantages of using customer programs:-.....	12
Disadvantage of using customer programs:-.....	12
Related Content.....	13
Disclaimer and Liability Notice.....	14

## Purpose

The basic purpose of this document is to provide guidelines which need to be followed in any data migration project. Data Migration is the most important aspect in any SAP implementation project. Incorrectly loaded master data can pose a serious threat to business and can also delay the implementation cycle from several days to several weeks.

The points mentioned in this article are generic points which need to be followed for any versions of SAP. The guidelines are not concrete steps which have to be followed religiously. They serve as measures against possible pitfalls.

## Techniques

There are two methods to be followed for data migration.

- LSMW
- Customer Developed Programs.

Each method has its advantages and disadvantages which are listed further.

## Procedure

The following course of action should be followed for data migration.

- Identifications
- Decision
- Development
- Testing
- Final Migration

Each of the action points can be described in details as follows.

### Identifications

The first step primarily lies in identifying the master data to be loaded. This includes:

- Identify the various master data that needs to be migrated.
- Identify the fields from legacy that need to be updated into SAP.
- Identify the various tables in SAP for the corresponding masters.
- Chalk out the exact field in SAP and map it to the corresponding legacy field.
- Identify the additional fields which are not present in SAP but exist in Legacy systems and have business importance and therefore need to be transferred to SAP.
- Identify the mandatory fields as per business needs.
- Identify the various methods available to load the particular master data in the 2 techniques.
- In case no standard program or IDOC or BAPI is available; then identify the BDC transaction to load the master data.
- In case no BAPI is available; identify if any standard function module is available to load the particular data. This could be useful in case where creating a BDC would be complicated and custom programs have to be built.
- Identify other than standard masters such as customer master, material master etc are there any other master data or business transaction which needs to be maintained in custom Y/Z tables.
- Identify the various customizing data that needs to be set up before any of the masters can be loaded.

- Identify standard transactions or methods or programs available for mass deletion of master data. This is needed incase erroneous data gets loaded into system and which needs to be removed.
- Identify whether the data contains special characters such as “É”. This needs to be handled with Unicode Ports in case of IDOCS.

## Decision

Decision involves analyzing the legacy data and deciding upon the various strategies.

- Decide upon the actual master data needed to be uploaded.
- Decide the exact fields from Legacy that need to be mapped to the fields in SAP.
- Decide upon the strategy and techniques to be followed. This would involve understanding the pros and cons of both the techniques and then deciding upon the technique to be used. LSMWs have an edge over Customer Developed Programs due to the fact that they are easier to develop and no need of transport requests. So whenever possible go for LSMWs.
- Decide upon the fields for storing legacy key fields incase of material master, customer master, vendor master etc. incase of internal numbering system.
- Decide upon the file format of the legacy data. Using tab delimited file for data upload is a good practice as it can be easily opened in a Spreadsheet such as Microsoft Excel or Open Office Spreadsheet for modifying erroneous data. The thing to be taken care here is that the initial zeros could be lost while opening from tab delimited file into a spreadsheet. So it could be better using spreadsheets as legacy files and then saving them into tab delimited file while loading the data.
- Decide upon the date format, time format and the decimal notation which needs to be followed. All legacy data coming from the legacy files should have the same decimal notation and date format. In most fields the date format expected is YYYYMMDD, time format as HHMMSS and the decimal separator as ‘.’ (Dot). This is the internal format for storing date, time and numerical data.
- Arrive at a common consensus about the date, time and decimal format and check that all users who are going to load the data have the same formats set in the SAP systems. This is very important incase data is to be loaded for different countries. A mess up with the decimal notation can cause a large impact on the business data.
- Decide upon using the initial zeros for certain data such as material or customer or vendor etc either in legacy files or make use of the conversion routine function modules in the LSMW or customer programs.
- Decide the sequence in which the master data is to be loaded. For example the Bill of Material cannot be loaded unless the Material Master is loaded or that the Bank Master needs to be loaded earlier of the Customer Master and / or Vendor Master.
- Decide the customizing settings that need to be carried out before and after of every data load.
- Decide the customer tables that need to be created for additional master or business data from legacy systems that cannot be mapped to any of the standard available tables.
- Decide upon the technique to be followed to upload data in the customer tables.
- Incase of customer programs to be built for data migration decide on the error and /or success output report that is needed after loading of the data.
- Decide upon the various business rules needed for filling up different fields as required.

## Development

Development involves designing the LSMW and/or Customer Programs for the data load.

- Prepare the file format of the legacy data.
- Classify the files into header and line item files as needed by the technique to be used.
- A file format could hold the following information.

	A	B	C	D	E	F	G	H	I	J	K
1	<b>Material Master - Basic Data</b>										
2	<b>Field Name</b>	<b>MATNR</b>	<b>MBRSH</b>	<b>MTART</b>	<b>MEINS</b>	<b>MAKTX</b>	<b>MATKL</b>	<b>BISMT</b>	<b>NTGEW</b>	<b>GEWEI</b>	<b>BRGEW</b>
3	<b>Field description</b>	Material number	Industry sector	Material type	Base unit of measure	Material description	Material group	Old material number	Net weight	Weight Unit	Gross Weight
4		Initial	Initial	Initial	Basic Data 1	Basic Data 1	Basic Data 1	Basic Data 1	Basic Data 1	Basic Data 1	Basic Data 1
5	<b>Table Name</b>	MARA	MARA	MARA	MARA	MAKT	MARA	MARA	MARA	MARA	MARA
6	<b>Field Name</b>	MATNR	MBRSH	MTART	MEINS	MAKTX	MATKL	BISMT	NTGEW	GEWEI	BRGEW
7	<b>Data type</b>	CHAR	CHAR	CHAR	UNIT	CHAR	CHAR	CHAR	QUAN	UNIT	QUAN
8	<b>Length</b>	18	1	4	3	40	9	18	13	3	13
9	<b>Decimal places</b>								3		3
10	<b>Mandatory</b>	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory	Optional	Optional	Optional	Optional

- Identify the mandatory fields which are needed in the BAPI FM or IDOC or Direct Input program or the BDC. These fields may not be mandatory from business point of view but needed for the BAPI Function Module or IDOC.
- Identify the fields which have a certain fixed values always and use them as constants in your LSMW or program rather than in the file from the legacy system.
- Put checks on legacy master data such as plant, sales organization, division etc against the check tables in the LSMW or customer programs so that incorrect data can be identified even before the actual loading of the data.
- Determine the various conversion routines which need to be utilized when converting data into the file to be loaded and use the same.
- In case of LSMWs in the field mapping step after putting the checks for the various fields use SKIP\_TRANSACTION to skip the record in case there is any error. Specify the record number and/or key fields so as to locate the corresponding record in the legacy file. Use colors to display the error messages.

### LSM Workbench: Change Field Mapping and Conversion Rules

```

PROUD-KERNEL - F2M - ASSET_MASTER Asset master
Code: * __GLOBAL_DATA__
DATA: gv_recono TYPE sytabix VALUE '1',
      gv_itm_cnt type sytabix.
---
BEGIN_OF_PROCESSING_ Before Data Processing Starts
BEGIN_OF_TRANSACTION_ Before Transaction Processing Starts
BALTD Interface for Legacy Data Transfer - Asset Accounting
---
Fields
---
BEGIN_OF_RECORD Before Using Conversion Rules
Rule : Default Settings Modified
Code: baltd = init_baltd.
      CLEAR: gv_itm_cnt .
      LOOP AT asset_item
          WHERE rechr_000001 =
              zasset-rechr_000001.
          ADD 1 TO gv_itm_cnt .
      ENDLOOP.
---
MANDT Client (Legacy Data Transfer FI-AA)
Rule : Default Settings
Code: BALTD-MANDT = SY-MANDT.
---
BUKRS Company Code
Source: ZASSET-BUKRS (Company Code)
Code: IF NOT zasset-bukrs IS INITIAL.
      DATA: lv_bukrs TYPE bukrs.
      SELECT SINGLE bukrs FROM t001
      INTO lv_bukrs
      WHERE bukrs = zasset-bukrs .
      IF sy-subrc NE 0.
          FORMAT COLOR COL_NEGATIVE.
          WRITE:/ 'For record no', gv_recono,
                'The company code', zasset-bukrs,
                ' is not valid'.
          FORMAT COLOR OFF.
          skip_transaction.
      ELSE.
          baltd-bukrs = zasset-bukrs.
      ENDIF.
      ELSE.
          FORMAT COLOR COL_NEGATIVE.
          WRITE:/ 'For record no', gv_recono,
                'The company code is mandatory'.
          FORMAT COLOR OFF.
          skip_transaction.
      ENDIF.
---
ANLKL Asset Class
Source: ZASSET-ANLKL (Asset Class)
    
```

- In case of customer programs it would be good to pass the actual data along with the error message concatenated in the end to the error file so as to identify the error record easily.
- Build up the proper configurations settings needed for IDOCS and BAPI in the LSMW settings as shown.

Project Selection	
Project	PROUD-KERNEL <input type="button" value="Conversion Objects"/>
Subproject	F2M Finance to Manage (FI-CO)
Object	ASSET_MASTER Asset master

## IDoc Inbound Processing: Preparatory Measures

Project

**IDoc Inbound Processing**

File port

tRFC port

Partn.Type

Partner No.

- Check whether the partner profiles are maintained correctly as needed in transaction WE20.

### Partner profiles

Partner  LSMW

Partn.Type  Logical system

**Post processing: permitted agent** **Classification**

Ty.  User

Agent  MREUZENA

Lang.  English

Outbound parmtrs.

Partner Role	Message Type	Message var	MessageFu	Test

Inbound parmtrs.

Partner Role	Message Type	Message var	MessageFu	Test
	ACC_DOCUMENT			
	BANK_CREATE			
	BATMAS			
	CAP_ACTOUT			

- In case of customer programs to be used for data upload; design a suitable interface for upload of different kinds of file from presentation and/or application server.
- Design the proper output report needed after loading of data.

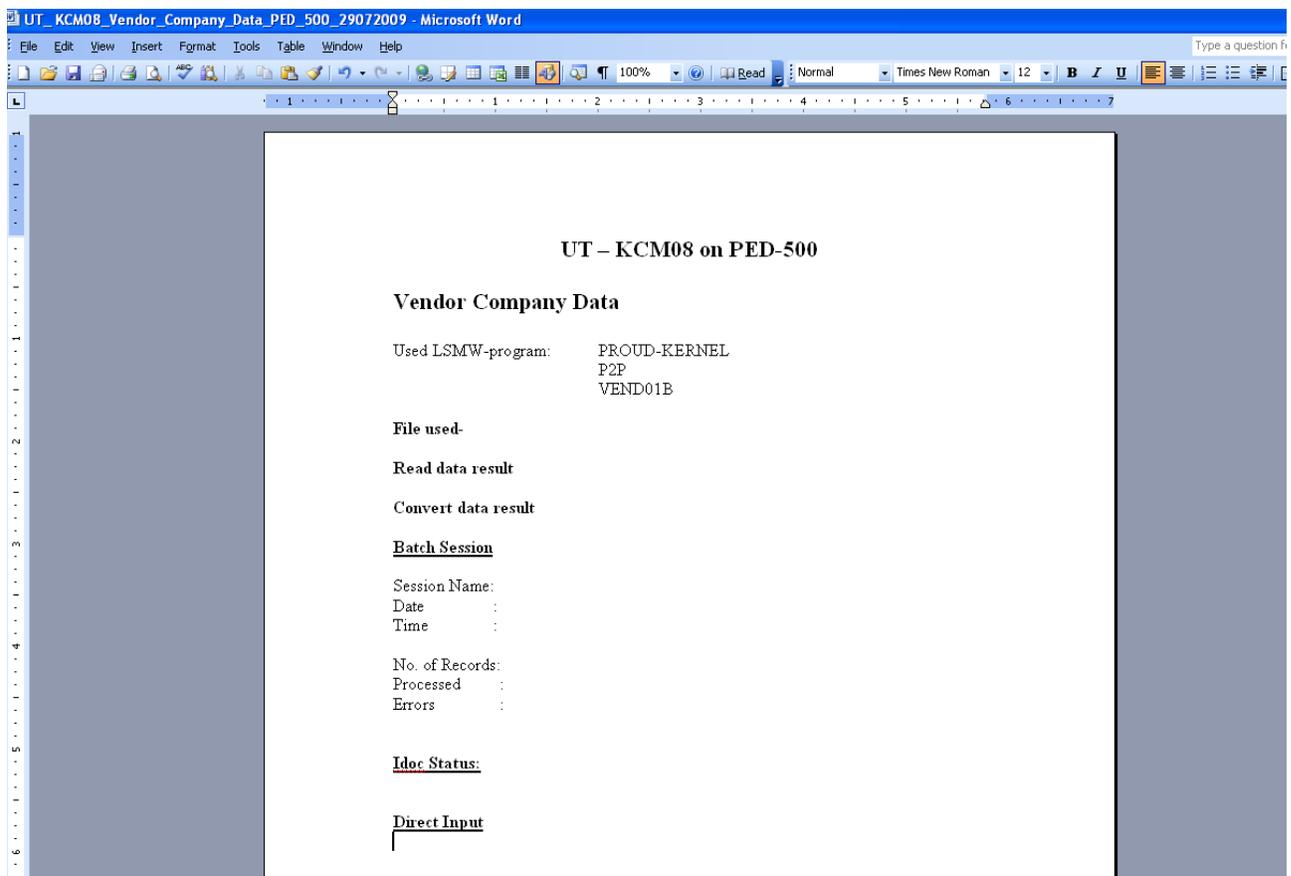
- In case of BDC sessions to be created via customer programs; provide the alternative of creating the session and processing of the same separately. RSBDCSUB is the standard SAP program available which can be used to process a session directly.
- In LSMW, handle the complex header item relationship in the Field Mapping step.
- Develop common subroutines which can be used across several LSMW sub objects or common subroutine pools in case of customer programs.
- Use the different business rules decided earlier so as to map the various fields.
- Put several rules so as to avoid loading of duplicate data or using duplicate files. For example; check if the material already exists before loading the material master and report error message if it does.

## Testing

Testing involves initial testing, load testing and integration testing.

- Before beginning with testing check if the customizing settings needed for the corresponding master are complete.
- Check if the user settings of date and decimal notation are as decided earlier.
- Before loading the file check the legacy file structure with the source structure of the LSMW or program to check for the field sequence and/or missing fields.
- Check if the mandatory fields are present in the legacy files and that those contain the relevant data.
- Check that the date, time and decimal format is followed in the legacy files.
- Check whether special characters used in several languages are present in the relevant data in the legacy file.
- In case of duplicate records; identify and report the same before moving with the process. In case of huge data loads if there are several duplicate records the process may take a longer time resulting in loss of valuable time needed during actual migration.
- Test the LSMW and program first with few records so as to check if there are no errors identified while loading the data.
- In case of LSMWs; after reading the legacy data, check few of the read records so as to check whether the data has been read properly.
- Check whether the number of records in the file and the number of records read is same. In case not debug the read program to identify the mismatch. A common mistake in this case is duplicate records or incorrect combination of header and item data.
- After the convert step check whether the data is converted properly.
- Check if the special characters in the data are converted properly and do not result in '#'.
- Check if the date format and decimal notation of the converted data is correct.
- Check if the converted data follows the various conversion routines used.
- Check if the converted data is as per the business rules used in the conversion logic.
- In case data is not converted fully; check if it is possible to correct the erroneous records in the file and then use the same again. At this step please take care of the tab delimited files opened in spreadsheets do not lose the initial zeros.
- After successful running of the load program; check if the data is correctly loaded in the relevant tables. Also check via the corresponding transaction of the data is displayed properly. There may be issues in case of special characters if the Unicode conversion is not followed properly.
- In case of any changes done in the programs or LSMWs after identifying the issues; check to it that the file structure is updated accordingly for the legacy system.

- After initial testing for few records; do testing with considerable amount of different kind of master data to check if the data is transferred properly. This is needed incase of several item records for a header records to check if the step loop is handled properly incase of a BDC.
- Check if the data loaded is correct in a business way. This would be on part of functional consultants and/or end users.
- Finally test for load testing with huge amounts of data. Do the load testing to check the possibility of time outs if any. Incase there are time outs a decision has to be taken as to handle considerable amounts of data in multiple files.
- Check if the mass deletion programs are functioning as needed so as to remove mass data incase of any issues.
- During integration testing; note the time required for loading each and every master data. This is with the assumption that in integration testing the same data would be loaded as the production data. The time factor will help to schedule the different data load and for planning simultaneous load of different masters.
- After completion of data loading; prepare a report for each file with details such as the input files, the converted records, the session details, the output report, the error records etc. An example is as shown



This is just a template which can be adjusted as needed.

- Incase of IDOCS; identify the error IDOCS and corresponding data passed so as to do a mapping to the erroneous records and report the same in the error report.
- Incase of sessions you can print the session data and then download it to a spreadsheet for proper analysis.

Log System Help

Batch Input Log for Session **GL\_BA** | Print from ABAP list viewer (ALV) (Ctrl+P)

Choose

Log attributes

Name: **GL\_BALANCES** Queue ID: **09091611525768061892** User: **KSHAH**  
 Created On: **09/16/2009** TemSe ID: **BDCL6576806189296258**  Details

Time	Message	Transact	Index	Modul	Scr	In	T	M	
11:53:07	Session GL_BALANCES is being processed by user KSHAH in mode A or		0			0	S	00	300
11:53:10	G/L account 15250000 CA10 does not exist	FB01	1	SAPMF05A	0100	1	E	F5	106
11:53:15	Field BDC_OKCODE changed: ?HC (previously: /00)	FB01	1	SAPMF05A	0100	1	S	00	297
11:53:28	Field BDC_OKCODE changed: =PICK (previously: )	FB01	1	SAPMSSY0	0120	1	S	00	297
11:53:28	Field RF05A-NEWKO changed: 10000000 (previously: 15250000)	FB01	1	SAPMF05A	0100	1	S	00	297
11:53:28	Field BDC_OKCODE changed: /00 (previously: ?HC)	FB01	1	SAPMF05A	0100	1	S	00	297
11:53:29	Enter another transaction type (Transaction type 702 does not exist)	FB01	1	SAPMF05A	0100	1	E	AA	816
11:53:55	Field BDC_OKCODE changed: ?HC (previously: /00)	FB01	1	SAPMF05A	0100	1	S	00	297
11:54:07	Field RF05A-NEWBW changed: 700 (previously: 702)	FB01	1	SAPMF05A	0100	1	S	00	297
11:54:07	Field BDC_OKCODE changed: /00 (previously: ?HC)	FB01	1	SAPMF05A	0100	1	S	00	297
11:54:22	G/L account 15450000 CA10 does not exist	FB01	1	SAPMF05A	0330	4	E	F5	106
11:54:38	Field BDC_OKCODE changed: /BEND (previously: /00)	FB01	1	SAPMF05A	0330	4	S	00	297
11:54:38	Batch input processing cancelled		1			4	A	00	383
11:54:38	Transaction error	FB01	1			0	S	00	357
11:54:38	Processing statistics		0			0	S	00	370
11:54:38	1 transactions read		0			0	S	00	363
11:54:38	0 transactions processed		0			0	S	00	364
11:54:38	1 transactions with errors		0			0	S	00	365
11:54:38	0 transactions deleted		0			0	S	00	366
11:54:38	Batch input processing ended		0			0	S	00	382

List Edit Goto Settings System Help

Batch Input Log for Session **GL\_BALANCES**

Local file... (Ctrl+Shift+F9)

Log time	Log message	Trans.	Index	Modules	Scr.	Ind	T	ID	No.
11:53:07	Session GL_BALANCES is being processed by user KSHAH in mode A on server prosappcdi						S	00	300
11:53:10	G/L account 15250000 CA10 does not exist	FB01	1	SAPMF05A	0100	1	E	F5	106
11:53:15	Field BDC_OKCODE changed: ?HC (previously: /00)	FB01	1	SAPMF05A	0100	1	S	00	297
11:53:28	Field BDC_OKCODE changed: =PICK (previously: )	FB01	1	SAPMSSY0	0120	1	S	00	297
11:53:28	Field RF05A-NEWKO changed: 10000000 (previously: 15250000)	FB01	1	SAPMF05A	0100	1	S	00	297
11:53:28	Field BDC_OKCODE changed: /00 (previously: ?HC)	FB01	1	SAPMF05A	0100	1	S	00	297
11:53:29	Enter another transaction type (Transaction type 702 does not exist)	FB01	1	SAPMF05A	0100	1	E	AA	816
11:53:55	Field BDC_OKCODE changed: ?HC (previously: /00)	FB01	1	SAPMF05A	0100	1	S	00	297
11:54:07	Field RF05A-NEWBW changed: 700 (previously: 702)	FB01	1	SAPMF05A	0100	1	S	00	297
11:54:07	Field BDC_OKCODE changed: /00 (previously: ?HC)	FB01	1	SAPMF05A	0100	1	S	00	297
11:54:22	G/L account 15450000 CA10 does not exist	FB01	1	SAPMF05A	0330	4	E	F5	106
11:54:38	Field BDC_OKCODE changed: /BEND (previously: /00)	FB01	1	SAPMF05A	0330	4	S	00	297
11:54:38	Batch input processing cancelled		1			4	A	00	383
11:54:38	Transaction error	FB01	1			0	S	00	357
11:54:38	Processing statistics		0			0	S	00	370
11:54:38	1 transactions read		0			0	S	00	363
11:54:38	0 transactions processed		0			0	S	00	364
11:54:38	1 transactions with errors		0			0	S	00	365
11:54:38	0 transactions deleted		0			0	S	00	366
11:54:38	Batch input processing ended		0			0	S	00	382

## Final Migration

Final Migration needs sufficient time for loading the data in the production system and then checking the same.

- Before loading the final data follow all the checks as mentioned in the testing step.
- Do not load the data unless all the customizing is complete.
- In case of LSMW; run the session or process the IDOCS or the direct input program only after all the errors identified in the READ step and CONVERT step have been resolved.
- Do not load the data unless all the identified errors have been resolved.
- Load few records of the final data first to identify any possible errors in the session or IDOC or direct input program. In case any error identified then check the legacy file for a similar kind of error in the remaining data and then load the actual final data.
- In case of huge records it would be advisable to schedule in background rather than online run.
- In case of wrong data uploaded run the corresponding mass deletion program to remove the erroneous records and then load with fresh data.
- In case of few errors in the loaded data; it would be advisable to make changes manually via the corresponding online transaction.

## Advantages / Disadvantages

### Advantages of using LSMW:-

- Easy to design and develop.
- Less time involved in design, development and maintenance.
- The output reports obtained are standard and therefore not needed to be designed.
- Upload of data from various formats of input files supported such as tab delimited, comma separated etc.
- Can be exported and imported to different systems without the need of transport requests.
- Can be rectified in any client.
- Good interface available to inspect read data and converted data so as to avoid errors before uploading.

### Disadvantage of using LSMW:-

- Complex coding needed for handling complex header and item relationship cannot be managed in the field mapping step.
- Not suitable in certain cases of table control where handling step-loop is difficult to code in the field mapping step.

#### Advantages of using customer programs:-

- Complex coding needed for handling complex header and item relationship can be managed efficiently.
- Suitable in certain cases of table control where handling step-loop is difficult to code in the field mapping step.
- Output reporting can be as per user designs.

#### Disadvantage of using customer programs:-

- Requires extensive coding to match all the advantages of using LSMW.
- The conversion logic and mapping of source fields to target fields need extensive coding.
- Although upload of data from various formats of input files supported such as tab delimited, comma separated etc it requires extensive coding for developing the same.
- Cannot be modified directly in different systems. Need to be transported every time for changes made.
- No interface available to inspect read data and converted data so as to avoid errors before uploading.
- Considerable amount of time and development efforts needed in design, development and maintenance.

Finally to summarize, the above said points should be considered before starting off with migrating data to the production systems. There are several other factors which need to be taken into account as per business needs but following the points will give a structured approach for carrying out data migration. This structured approach will always help to avoid possible pitfalls during the final migration.

## Related Content

[http://help.sap.com/saphelp\\_nw04/helpdata/en/ad/2d54a41d7011d2b42e006094b944c8/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/ad/2d54a41d7011d2b42e006094b944c8/frameset.htm)

[http://help.sap.com/saphelp\\_nw04/helpdata/en/6f/c7b041548011d192fd0000e829fbc6/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/6f/c7b041548011d192fd0000e829fbc6/frameset.htm)

For more information, visit the [ABAP homepage](#).

## 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.