



# **Best Practice Guide – Classical Migration of SAP NetWeaver Application Server ABAP to SAP HANA**

Version from July 07, 2016

# TABLE OF CONTENTS

<b>1</b>	<b><i>Introduction</i></b>	<b>3</b>
<b>2</b>	<b><i>Considerations Before the Migration</i></b>	<b>4</b>
2.1	Important Information about the Migration to SAP HANA	4
2.2	Prerequisites and Restrictions	4
2.3	Involved Tools	4
2.4	Considerations for Performance	6
2.5	Preparations	7
<b>3</b>	<b><i>Considerations During the Migration</i></b>	<b>12</b>
3.1	Export	12
3.2	Transfer of Export File	13
3.3	Import	13
<b>4</b>	<b><i>Considerations After the Migration</i></b>	<b>16</b>
4.1	Consistency Checks	16
4.2	Post-Processing	16
4.3	Troubleshooting	16
<b>5</b>	<b><i>Appendix</i></b>	<b>18</b>
5.1	Example – Optimizing the Runtime for Migration of SAP BW Test System	18

© 2016 SAP SE. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP SE and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

## 1 INTRODUCTION

This document provides **general expert and best practice information especially for the classical migration to SAP HANA** (that is, heterogeneous system copy using classical migration tools, such as *software provisioning manager 1.0* and *R3load*).



### NOTE

Find always the latest version of this document in SAP Community Network on the *System Copy and Migration* page at <http://scn.sap.com/docs/DOC-8324> (direct link: <http://scn.sap.com/docs/DOC-47657>). There, you can also provide feedback, which helps us to improve the guide further. Please also rate the document, if you find it useful.

- Target group of this documentation are technical consultants experienced with the migration.
- While the main focus of this document is on SAP NetWeaver 7.4, it is also relevant for SAP NetWeaver 7.3x, powered by SAP HANA (including SAP Business Warehouse).
- Although some aspects might also be relevant for the migration to other databases, the corresponding best practices were only validated for the migration to SAP HANA.
- Migration is customer- and system-specific – use the best practice recommendations from this section with caution.



### NOTE

This guide does not replace the system copy documentation, but complements it. For detailed information about migration in general, refer to the **SAP HANA System Copy Guide** available in SAP Help Portal at: <http://help.sap.com/sltoolset> → *Area System Provisioning*.



### NOTE

For an overview of migration path options to SAP NetWeaver 7.4 running on SAP HANA, see the End-to-End Implementation Roadmap guides available at: <http://help.sap.com/nw74/#section7>

- For SAP NetWeaver Application Server ABAP 7.4 running on SAP HANA, see the [End-to-End Implementation Roadmap for SAP NetWeaver AS ABAP](#).
- For SAP Business Warehouse (SAP BW) running on SAP HANA, see the [End-to-End Implementation Roadmap for SAP BW](#).

To migrate your source database to the SAP HANA database, you perform a heterogeneous system copy using standard tools, such as *software provisioning manager*, *SAPinst*, *R3load*, and *Migration Monitor*.

A heterogeneous system copy has the following characteristics:

- Operating system is changed (in this case, system copy is called OS migration – not relevant for the migration to SAP HANA) or
- Database system is changed (in this case, system copy is called DB migration) or
- Operating system and database system are changed (in this case, system copy is called OS/DB migration)

Heterogeneous system copy is performed using a database-independent procedure (with some exceptions for certain OS migrations, where cross-platform backup or restore is supported).

## 2 CONSIDERATIONS BEFORE THE MIGRATION

### 2.1 Important Information about the Migration to SAP HANA

#### 2.1.1 Generic Information

Read the following information:

- [Blog](#) from Stefan Seemann in SAP Community Network about R3\* tools provided with the SAP kernel that are involved in the migration to SAP HANA (*Migration to SAP HANA: SAP Kernel Update for the Migration*)
- In addition, take a look at the blog [Migration to SAP HANA using Software Provisioning Manager: How to Begin?](#).
- [Blog](#) from Sunny Pahuja providing further details for the migration to SAP HANA – with focus on SAP BW, but partly also valid for general migrations to SAP HANA.

#### 2.1.2 Application-Specific Information

For the migration of SAP Business Warehouse (SAP BW), see the following information in the SAP Community Network:

- [SAP BW Application Lifecycle Management](#)
- [Best practice information for SAP BW powered by SAP HANA scale out](#)

### 2.2 Prerequisites and Restrictions

- Delivery of SAP HANA appliances can take some time - therefore, order required target hardware early.
- Before starting the database migration, perform a hardware check using the script `HanaHwCheck.py` – for more information, see SAP Note [1652078](#) (*SAP HANA database: Hardware check for more details*). In case the hardware check fails, try the latest version of the script that is attached to SAP Note [1658845](#) (*Recently certified SAP HANA hardware not recognized*).
- For recommended Linux settings, refer to SAP Note [1824819](#) (*SAP HANA DB: Recommended OS settings for SLES11/SLES4SAP SP2*).
- Install the latest available revision of SAP HANA before the migration.
- Make sure that you install a current SAP HANA client that has the same version as the SAP HANA database. For more information about how to install a current SAP HANA client, see SAP Note [1825053](#) (*Installation of current SAP HANA client with SWPM*).
- Install the SAP HANA studio as described in the *SAP HANA Studio Installation Guide* available at [http://help.sap.com/hana\\_appliance](http://help.sap.com/hana_appliance) → *Installation and Upgrade Information* and in SAP Note [1789632](#).
- Installation of external software on the HANA appliance must be in compliance with SAP Note [1730928](#) (*Using external software in a HANA appliance*).
- To prepare the adjustment of your custom code, install the ABAP Development Tools as described in the guide provided in SAP Note [1718399](#).
- For SAP Business Warehouse, check SAP Note [1600929](#), providing further information relevant for the migration to SAP HANA.

### 2.3 Involved Tools

This section provides expert information and recommendation for some of the involved migration tools. For an overview of the most important migration tools for ABAP, see <http://scn.sap.com/docs/DOC-34258>.

#### 2.3.1 Software Provisioning Manager

Software provisioning manager 1.0 offers many services, including system installation and system copy. For our use case of database migration, it orchestrates all involved tools.

Especially for SAP HANA, customers can fully benefit from the latest corrections regarding performance, robustness, and scale-out through a faster delivery cycle option in *software provisioning manager*. Latest versions continue to be made available via the *Software Logistics Toolset*, as outlined below.

This accelerated delivery option is especially important for customers performing a classical migrating of ABAP systems to SAP HANA – for more information about latest corrections in software provisioning manager for the migration to SAP HANA, see the corresponding blog [Migration to SAP HANA: Latest News about Software Provisioning Manager 1.0](#) in SAP Community Network.

### 2.3.2 Migration Monitor (Migmon)

*Migration Monitor* is part of *software provisioning manager*. It uses the `EXT`, `WHR`, `STR`, `TPL` files to control the unload and load, to accelerate the load by automatic parallelization, to trigger the creation of the *R3load* task files, and creates the command files (`TSK` and `CMD` files). *Migration Monitor* does the following:

- Creates *R3load* command files
- Triggers creation of *R3load* task files if required (created by *R3load*)
- Starts *R3load* processes to unload the data
- Transfers packages from source to target host if required
- Starts *R3load* processes to load data as soon as a package is available
- Informs the person performing the system copy in case of errors

As of SAP NetWeaver 2004 SR1, the *Migration Monitor* is integrated in *SAPinst/software provisioning manager*, but it is also possible to start *Migration Monitor* manually (there is a corresponding option in the *software provisioning manager*) – for this, you require a properties file. When you start the *Migration Monitor* manually:

- You can adjust any parameters according to the *Migmon* user guide
- You get flexible (for example, you can repeat, test and abort runs of the *Migration Monitor*)
- The process gets more complex, as it requires many additional manual activities
- The properties file has to be created manually (hint: re-use an existing properties file from previous runs of *software provisioning manager* as template)

Especially for SAP HANA, the *Migration Monitor* comprises a `migmonctrl.jar` add-on that gets invoked automatically to adjust the amount of *R3load* jobs dynamically during the import – this is described in the *Import* section below.

For more information, see:

- The *Migration Monitor User's Guide* explains parameters, functions, and control files in more detail. The guide can be extracted from the archive `MIGMON.SAR` in the *software provisioning manager* package.
- SAP Note [784118](#) (*System Copy Tools for ABAP Systems*)
- Section *Restarting R3load Processes* in the *System Copy Guide*.

### 2.3.3 R3load

*R3load* performs all load tasks in a database- and platform-independent format. It generates the database export of all SAP objects that are defined in the ABAP Dictionary, including the configuration and components in the file system. Then, it performs the load of ABAP tables into the target database. It is available via the kernel of the ABAP application server.

### 2.3.4 R3ldctl

*R3ldctl* makes information of source system database tables available for migration tools offline by reading ABAP Dictionary structures from the source database. It creates structure (`STR`) files that describe the definition of tables, indexes and views, and it creates database-specific template (`TPL`) files containing commands in Data Definition Language (`DDL`), which define data structures. For more information, see the *System Copy Guide* → *Exporting the Source System Using the Installer*. *R3ldctl* is also available via the kernel of the ABAP application server.

### 2.3.5 R3ta

To copy large tables, use the *R3ta* table splitter tool – it automatically generates `WHERE` conditions with which a subset of table data can be accessed and with which the export and import of one table can be performed in parallel by starting several *R3load* processes that work on packages. These `WHERE` conditions are

integrated into the *R3load* TSK files. If the export or import of a table is cancelled with an error, only the processing of the package with the error has to be repeated instead of loading or unloading the complete table another time. *R3ta* is available via the kernel of the ABAP application server.

### 2.3.6 Package Splitter

*Package Splitter* reduces the overall runtime of export and import by splitting default packages into smaller parts using the existing structure files (STR + EXT). It is part of *software provisioning manager*.

### 2.3.7 Time Analyzer (Migtime)

To get statistical data about runtimes, use *Migtime*. It supports you in analyzing the runtimes of the export and the import:

- It calculates the runtime per package
- It lists the long running objects within a package for further splitting
- It creates a list (either a text or an HTML file) with start and end date or time per package
- It is provided with *software provisioning manager 1.0* (MIGTIME.SAR is available in <Extracted Software Provisioning Manager Archive>\COMMON\INSTALL)

For more information, see SAP Note [784118](#).

### 2.3.8 Tools Specific for SAP BW

For SAP BW, consider to use the following specific tools:

- A specific checklist tool for SAP BW (see SAP Note [1729988](#) – *SAP BW powered by SAP HANA - Checklist Tool*) automates the check of best practice guidelines for operations and pre-requisites for the migration of an existing SAP BW system to SAP HANA.
- Before migrating to SAP BW on SAP HANA, you can use the SAP BW ABAP Routine Analyzer to identify ABAP statements that can potentially be optimized for SAP HANA – for more information, see SAP Note [1847431](#) (*SAP BW ABAP Routine Analyzer*).
- To make the migration of an existing SAP BW deployment to SAP HANA easier, the SAP NetWeaver Migration Cockpit for SAP HANA combines several tools – for more information, see SAP Note [1909597](#) (*SAP BW Migration Cockpit for SAP HANA*).

## 2.4 Considerations for Performance

Note the following recommendations for the performance of the migration procedure:

- As a rule of thumb, it is recommended that you configure two or three parallel R3load processes per CPU on the host on which the corresponding R3load is running.
- Closely monitor the CPU and I/O utilization to adapt the number to actual project conditions.
- If possible, keep a dedicated internal disk with an own controller for the export directory.
- Using NFS-mounted file systems: Bear in mind that the overall performance of the system copy depends on all links in the chain: Starting from the performance of the source database,
  - To the performance of the server on which the export is executed
  - To the file system the export data is written to
  - To the target side which reads from the export medium and imports it into the target database.

You have to make sure that all chains are configured to deliver their best performance during the process. For recommendations on the corresponding Network File System (NFS) configuration, see SAP Note [2093132](#).

- Only change table type from column store to row store, if proposed by SAP HANA development support.
- We recommend to use the *massloader* option of *software provisioning manager 1.0*:
  - For this, SAP HANA Revision 51 or higher is required (see SAP Note [1806935](#))
  - To activate the option, set the environment variable `HDB_MASSIMPORT=YES` before starting *software provisioning manager 1.0*
- With the latest versions of *software provisioning manager 1.0*, parallel export/import is available for products based on SAP NetWeaver 7.3x. For latest information, see SAP Note [1775293](#).

- If scale-out is released for your product, consider to use an SAP HANA standby server node for the import to improve performance. For more information, see the corresponding information in section *Import* [page 13] below.
- Throughput is quite individual and depends on the hardware configuration and other factors. You can use the Java tool *Migtime* to evaluate the runtime of export/import – for more information, see SAP Note [784118](#). It might require multiple runs to find bottlenecks and optimize them.
- For comparing ABAP code execution times after a migration to SAP HANA, consider the recommendations in SAP Note [1942889](#) (*Comparing ABAP code execution times after HANA migration*).

## 2.5 Preparations

### 2.5.1 Unicode Conversion

SAP HANA requires Unicode. Should you have a non-Unicode system, perform a Unicode conversion before the migration or plan a combined migration with Unicode conversion, which is technically also possible. Be aware that R3load requires significantly more CPU time with a combined Unicode conversion, resulting in a reduced number of configurable R3load jobs. For more information, see the [Unicode Conversion Overview Guide](#) available in SAP Community Network.

*Distribution Monitor* can optimize the downtime of a Unicode conversion by distributing the R3load workload to several machines – nevertheless, *Distribution Monitor* is not released for SAP HANA, so it can't be used in a combined migration and Unicode conversion procedure, but only if you convert to Unicode as separate activity before the migration. Because the approach with *Distribution Monitor* increases the complexity of the conversion, use it only if really needed. For more information:

- SAP Note [855772](#) (*Distribution Monitor*)
- SAP Note [1001383](#) (*Distribution Monitor - Known problems and fixes*)
- User guide that is part of the tool archive (`DISTMON.SAR`)
- SAP Note [989116](#) (*Distribution Monitor: Troubleshooting Guide*)

### 2.5.2 Migration Tool Versions

Always use latest versions of the migration tools:

- Software provisioning manager:
  - You can get the latest version from <http://help.sap.com/sltoolset>
  - For more information, see SAP Note [1680045](#) (*Release Note for software provisioning manager 1.0*) and the blog [Migration to SAP HANA: Latest News about Software Provisioning Manager 1.0](#) in SAP Community Network.
- R3\* tools (`R3load`, `R3ldctl`, `R3szchk`, `R3ta`):
  - You can get the latest version from the tools from: <http://support.sap.com/swdc> → *Support Packages and Patches* → *A – Z Alphabetical List of Products* → *K* → *SAP KERNEL* → *<VERSION>* → *<Release>* → *<Platform>*.
  - For more information, see the blog [Migration to SAP HANA: SAP Kernel Update for the Migration](#) in SAP Community Network.
  - Verify that the versions are compatible. R3\* tools must fit the kernel version of your SAP system.
- SAPCAR:
  - You can get the latest version from the tools from: <http://support.sap.com/swdc> → *Support Packages and Patches* → *A – Z Alphabetical List of Products* → *S* → *SAPCAR*.

### 2.5.3 Kernel

Use the latest kernel media for *software provisioning manager* as listed in SAP Note [1680045](#) (*Release Note for software provisioning manager 1.0 → Kernel media for SWPM\**). Make sure to use the same kernel version both for export and import. For more information, see the *System Copy Guide → General Technical Preparations*.

### 2.5.4 Creating DDL Statements

Before you start the export, make sure to run the report `SMIGR_CREATE_DDL` that generates DDL statements for the migration:

1. Only valid for SAP NetWeaver 7.4 systems on SAP HANA (not for SAP BW 7.3x and SAP BW 7.4): If you migrate to a distributed SAP HANA database, make sure to also consider the prerequisites as listed in SAP Note [1781986](#) (*Business Suite on SAP HANA Scale Out*), such as running the report `SHDB_GROUP_TABLES_LOAD_BASED` to classify tables in the context of the migration. The report must be executed before the migration, in particular before running `SMIGR_CREATE_DDL`.
2. In addition, make sure that you applied all SAP Notes referenced in SAP Note [1921023](#) in the source system. Especially for migrations to SAP BW on SAP HANA, also apply the SAP Notes listed in the attachment `REQUIRED_CORRECTION_NOTES.TXT` of SAP Note [1908075](#). These SAP Notes contain corrections and enhancements to `SMIGR_CREATE_DDL`, such as:
  - Rowstore list:
    - For SAP NetWeaver 7.3x, the file `rowstorelist.txt` is used to distinguish between row-store and column-store packages, which can get created via an additional option of the report `SMIGR_CREATE_DDL`.
    - For SAP NetWeaver 7.4, corresponding information is stored in DDIC instead of `rowstorelist.txt`, so no action is required.
  - Table list with estimated record count:
    - `SMIGR_CREATE_DDL` creates a file `HDB_ESTIMATES.TXT` or `ESTIMATED_ROW_COUNT.TXT` that contains a list of all relevant tables and the estimated number of rows per table. This information is used during migration into scale-out systems after the initial table creation and before the actual load of data. At that point, the *software provisioning manager* performs an initial landscape redistribution of the empty tables. The estimated record counts are used in this step to ensure an equal distribution of data across the index server slave nodes.
    - Jochen Becker's [blog](#) in SAP Community Network provides additional information on table placement and landscape redistribution.
3. For migrations into scale-out systems:
  - Follow the recommendations provided in SAP Note [1908075](#) (set database parameters, maintain entries in control table for table placement, grant authorizations).
  - Check SAP Note [1958216](#) (*HANA landscape redistribution configuration*) for revision-specific parameter settings.
4. For partitioning, make sure that SAP Note [1783937](#) (*SMIGR\_CREATE\_DDL Enhancement for Suite on HANA*) is applied, so that partitioning restrictions are recognized when SQL files are created (that is, large tables get partitioned during the migration). Otherwise, the import of tables with more than 2 billion records (German: 2 Mrd.) will fail.
5. If you want to migrate an SAP Business Suite system to a distributed SAP HANA database, see SAP Note [1899817](#) (*Suite on distributed HANA database - table redistribution*).
6. Run the report `SMIGR_CREATE_DDL` to generate the DDL statements.

### 2.5.5 Table Splitting Preparation

For more information about how to perform table splitting, see section 4.3 *Preparing the Table Split* in the *System Copy Guide*.

#### Table Splitting

- For prerequisites, see SAP Note [1783927](#) (*Prerequisites for Table Splitting with target HANA database*).
- We recommend that you perform a test run to identify the tables with the longest runtime, as not always the largest tables require the longest runtime. To get statistical data about runtimes, use *Migtime* during the export and import (for more information, see SAP Note [784118](#)).
- Otherwise, create a list of the largest tables in the database before the export to identify candidates for table splitting (by using transaction `DBACOCKPIT` or `DB02`).
- It is recommended that you split tables to 10 million lines per table.



### *R3ta Tool*

For using *R3ta*, consider the following:

- *Software provisioning manager* offers a corresponding option in the export dialog – see below for more information.
- Under certain conditions, it is recommended that you create additional temporary indexes on the column used in the `WHERE` condition. Depending on the database, this may not be feasible during productive operation because of time and table locking.
- In contrast to previous statements, you can also use database-specific splitters, not only the generic *R3ta*.
- As for all R3\* tools, always use the latest version of *R3ta*. Using an outdated version of *R3ta* increases your risk of receiving incomplete `WHR` files, which can cause data corruption.
- If parallel processing is not optimal or if the single packages are serially processed, the complete processing time for one table can increase when using `WHERE` conditions.
- The export and import must be performed with *Migration Monitor* if the table splitting feature is used.
- Some platforms allow the creation and usage of an `R3ta_hints.txt` file, with which you can instruct *R3ta* to use specific columns of a table to create a `WHERE` condition. For more information, see the *System Copy Guide*, section *Preparing the Table Split → Using Hints*.
  - If you plan to use `R3ta_hints.txt`, check whether it contains a valid column for the table that you want to split. You can reduce the runtime of *R3ta* significantly by choosing a table column with a good selectivity. If no field is provided, a selectivity analysis has to be performed over several columns, which will increase the overall runtime of the splitting.
- Restrictions as listed in SAP Note [1738258](#) (*System Copy of Systems Based on SAP NetWeaver 7.3 and Higher*) → section *Table Splitting: Restrictions*:
  - The tables to be split must be listed in the input file in alphabetical order.
  - The following tables must not be split:
    - DDNTF
    - DDNTF\_CONV\_UC
    - DDNTT
    - DDNTT\_CONV\_UC
    - DDLOG
    - REPOLOAD

### **2.5.6 Package Splitting**

Package splitting is activated per default. Do not deactivate it.

*Package Splitter* can be used for splitting the following:

- `STR` (initially created by *R3ldctf*) and `EXT` files (provide sizes of tables, created by *R3szchk*)
- 6.40: `TSK` files with `WHERE` clauses (including an adaptation of the corresponding `STR` and `EXT` files)
- 7.00: `WHR` files with `WHERE` clauses (including an adaptation of the corresponding `STR` and `EXT` files)

*Package Splitter* (and *R3ta*) offers the following options that you can set in the corresponding export dialog of *software provisioning manager*.

### SAP System Split STR Files

Enter the parameters for the Java splitting tool.

**Splitting Tool Parameters**

Largest Tables in Separate Packages	<input checked="" type="checkbox"/>
Number of Tables to Be Extracted *	<input type="text" value="10"/>
Split Packages with Size More than Limit	<input checked="" type="checkbox"/>
Package Size Limit [MB] *	<input type="text" value="1000"/>
Split Packages by Number of Tables	<input checked="" type="checkbox"/>
Number of Tables Limit *	<input type="text" value="1000"/>
Extract Tables with Size More than Limit	<input type="checkbox"/>
Table Size Limit [MB] *	<input type="text" value="300"/>
Split Predefined Tables	<input checked="" type="checkbox"/>
Table Input File *	<input type="text" value="/sapdb/exportSDB/ABAP/DATA/whr.txt"/> <input type="button" value="Browse..."/>

**Additional Information**

If you want to split by *Number of Largest Tables*, by *Package Size Limit*, or by *Table Size Limit* (first three splitting options), EXT files are mandatory and have to be created during the export.

If EXT files do not exist for your database platform, the first three options will be ignored if selected and you have to specify an input file.

- **Number of tables to be extracted**  
Defines how many of the largest tables of each package will be extracted, default value is 10. As some packages might contain many tables (in the range of about 10.000 tables), you might want to increase this value (see values below).
- **Package Size Limit**  
This parameter defines the maximum size of an STR package, comprising all tables. In case the size should be above this limit, the package will be split. Default value is 1000MB.
- **Number of Tables Limit**  
This new split feature (available with the latest tool versions) runs after the other package splitting options finished. It takes care that every STR file contains the maximum amount of tables or less. This is helpful in case a package contains a lot of tables with less content (or empty tables – for example, in case of SAP BW migrations, where you might have a lot of empty tables in one STR file): while the other split options would not get active due to the small size, this split feature would help to reduce the corresponding runtimes.
- **Table Size Limit**  
This parameter defines the maximum table size – if a table should be larger, it gets extracted from the STR file. Default value is 300MB.
- **Split Predefined Tables**  
Optionally, specify a file containing tables (in most cases: your largest tables) that you want to get split into several pieces. As soon as the export of one split part has finished, the target side can already start processing (instead of having to wait for the export of the whole table).

Table splitting is done by *R3ta* (see above). For the other options, the *Package Splitter* is used by *software provisioning manager*.

The best way to find the optimum of these parameters is to run migration tests and afterwards analyze them using the *Migtime* tool – for more information, see SAP Note [784118](#). As a starting point, try the following parameters:

- Extract the 150 largest tables
- Extract tables larger as 200MB from the package
- Package size: 1000MB
- Corresponding command line (can also be triggered in corresponding dialog):  

```
> ./str_splitter.sh -strDirs -top 150 -tableLimit 500 -packageLimit 1000
```

For more information, see:

- SAP Note [784118](#): system copy tools for ABAP systems
- Section *Using the Package Splitter Tool* in the *System Copy Guide*.

### 2.5.7 Housekeeping

Before the migration, delete and archive obsolete data from tables that show the largest growth in data volume. For more information, see:

- The Data Management Guide for SAP Business Suite available in SAP Service Marketplace at: <http://service.sap.com/dvm> → *Additional Information* → *Data Volume Management*.
- SAP Note [679456](#) (*Reducing data volume before Unicode conversion*).

If not instructed otherwise, carry out the corresponding activities to reduce the data volume during system uptime.

Implement SAP Note [1659622](#) in the source system. It provides the ABAP report `SMIGR_BIG_ROW_STORE_TABS`. This report can be used to identify large tables that will be stored as row store tables in SAP HANA. A small memory footprint of the row store is beneficial to minimize the database startup time.

Especially for SAP BW, you can profit from an automated housekeeping, offered as task list of the ABAP task manager for lifecycle management automation. For more information, see SAP Note [1829728](#) (*BW Housekeeping Task List*).

### 2.5.8 Further Preparations

- Read SAP Note [1775293](#) (*Migration/system copy to SAP HANA using the latest SWPM 1.0*)



#### NOTE

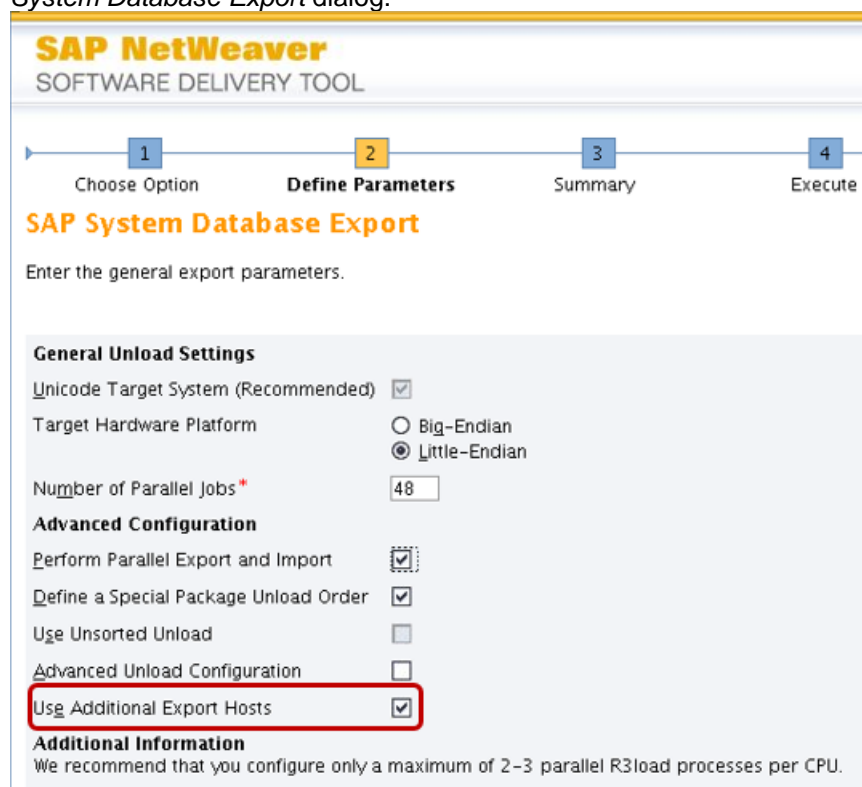
If you want to get automatic notifications about changes to this SAP Note, you can configure to receive email notifications on this [page](#) in SAP Community Network.

- `R3szchk` calculates space requirements on the target database for ABAP tables and indexes and rough estimation of overall size for target database. Per default, `R3szchk` gets executed by software provisioning manager. The result is an `EXT` and a `DBSIZE.XML` file. If `R3szchk` is very slow, check SAP Note [1047369](#).
- Request or generate a migration key. For more information, see <http://support.sap.com/migrationkey>.
- Before the export, run the reports `SDBI_CLUSTER_CHECK_PERFORM` and `SDBI_CHECK_BCD_NUMBERS`. This avoids issues during the export and import because of corrupt or initial cluster records or decimal numbers. For more information, see SAP Note [1784377](#) (*Checking pool tables and cluster tables*).
- Avoid more than one statistics server for your SAP HANA database. Otherwise, the import step can fail because of a 'standby' statistics server. Remove any configured standby statistics servers before the import.
- Perform a test run of the system copy procedure to calculate downtimes of your source system based on experiences made during the test run. Adapt the general process to your system and environment. For more information, see the *System Copy Guide*.
- Order and install a new SAP license. For more information, see the *System Copy Guide*.
- Prepare the source system as described in the *System Copy Guide*. For example, check for canceled or pending update requests and operation mode.

### 3 CONSIDERATIONS DURING THE MIGRATION

#### 3.1 Export

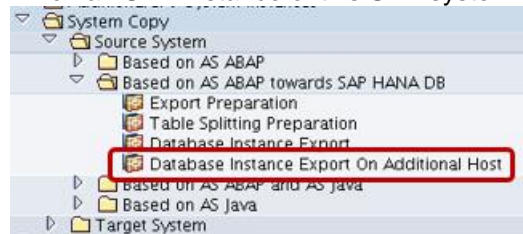
- Declustering is performed by default during the export – that is, cluster tables are made transparent. This is also the case if you are using parallel export/import with the Net Exchange or the FTP option. Only if you are using parallel export/import with the Socket Mode option, declustering has to take place during import. Even in this case, no direct action is required from you, as the latest version of *software provisioning manager/Migration Monitor* adapts the procedure automatically.
- With parallel export/import, SAPNTAB gets imported first (sequentially), before parallel import is started.
- Perform the splitting as prepared above under [Preparations](#) [page 7].
- Sorted vs. unsorted export:  
The default setting for the data export is *unsorted* (the default got changed from *sorted* to *unsorted* with only some hard-coded exceptions). It is recommended that you keep the defaults. Only change the default on explicit advice by development support.
- Performance-relevant considerations for the export:
  - Check if SAP Note [1775293](#) (*Migration/system copy to SAP HANA using latest SWPM 1.0*) contains any performance-relevant aspects.
  - Optionally, you can use more than one SAP application server to increase the performance of the export:
    - In Software Provisioning Manager, start also here with the option *Database Instance Export*, but select the “Use Additional Export Hosts” in the corresponding *SAP System Database Export* dialog.



Remark: it is not necessary to combine this feature with the execution of a parallel export and import (although this option is selected in the screenshot above).

With this, a package filter file is created for each host involved in the export – you can specify details in feature-specific dialogs that come up in the course of the dialog phase (such as how many package filter files/hosts you want to involve in the export, the name of the files, and the unload order). Per default, the same amount of packages (measured by size or runtime) is assigned to each single package filter file respective host.

- For each additional package filter file you have specified before, start Software Provisioning Manager with the option *Database Instance Export on Additional Host*, which exports the remaining packages, as defined in the package filter file. For this, you can start Software Provisioning Manager on the same host or any other host on which an SAP instance of this SAP system is running.



- Update database statistics, if the source database requires a manual refresh of the database statistics.

### 3.2 Transfer of Export File

Files can be transferred from the source host to the SAP HANA host using different protocols, for example NFS or FTP. In advance, consider the following aspects:

- Performance of the file transfer types
- Availability on different platforms (NFS, for example, is not available on Microsoft Windows hosts)
- Security requirements (for example, *Migration Monitor* currently only supports FTP, but not SFTP)
- Availability in a customer system landscape (FTP, for example, can be forbidden in customer landscapes)
- Which transfer mode and protocol is supported by *software provisioning manager* and *Migration Monitor* for the parallel export and import?

Which tool and process is in control of the overall export, transfer, and import process?

### 3.3 Import

#### 3.3.1 Declustering

If you are using parallel export/import with the Socket Mode option, declustering has to take place during import. For this, no direct action is required from you, as the latest version of *software provisioning manager*/*Migration Monitor* adapts the procedure automatically.

#### 3.3.2 Log Mode

Before the import, make sure that the log mode is set to *overwrite* (which is the initial value of the log mode after the initial setup of the database until a first data backup has been performed) to avoid a “disk full” situation.

In case data backups already exist for this instance, you can set the log mode back to *overwrite* by changing the parameter `log_mode` from `normal` to `overwrite` and restarting the database, so that the new parameter becomes active.

#### 3.3.3 Table Placement

For scale-out systems, download the relevant `TABLE_PLACEMENT` text file that is attached to an SAP Note. The relevant SAP Note for your application and SAP HANA revision is documented in SAP Note [1900822](#).

The `TABLE_PLACEMENT` text file should then be used as parameter file for *software provisioning manager* during the import. Jochen Becker's [blog](#) in SAP Community Network provides additional information on table placement and landscape redistribution.

### 3.3.4 Check Parameters for Dynamic Adjustment of Load Jobs

Especially for SAP HANA, the *Migration Monitor* comprises a `migmonctrl.jar` add-on that gets invoked automatically to adjust the amount of R3load jobs dynamically during the import. For this purpose, the add-on maintains the `ORDER_BY.TXT` file, which is read by *Migration Monitor* every 30 seconds to start new jobs. The adjustment is necessary because most of the tables need to be organized (merged) in memory before they can be flushed to the disk. Therefore, an additional amount of memory is necessary.

In general, all packages of an export are distributed over three groups inside the `ORDER_BY.TXT` file:

- `[Large]`  
Group *Large* contains the biggest packages sorted by size from biggest to smallest.
- `[Small]`  
Group *Small* contains the smaller packages sorted by size from smallest to biggest.  
As Migration Monitor starts the jobs according to the number of jobs defined for one group top-down, the first table of group *Large* (the biggest) is loaded together with the first of group *Small* (the smallest). Due to this, SAP HANA memory is used more efficiently.
- `[Rowstore]`  
This group contains packages that only contain one row-store table – that is, tables not listed here are column-store tables. Row-store tables always stay in the memory (also after import), whereas column-store tables are merged, compressed and synchronized to disk after import, which reduces memory consumption.  
The *Rowstore* group is loaded at the end of the import to avoid interferences with column-store tables.

The add-on has parameters that you can adjust in the `MIGMONCTRL_CMD.PROPERTIES` file. Check these parameters and if required, adjust them according to your situation. These are the most important parameters:

```
# Maximum number of R3load jobs defined in groups Large and Small.
maximumJobNum=100
# Factor of safety used to multiply the current size of all running jobs.
# The number of jobs is increased as long as enough free space is available
# within HANA.
memoryFactor=2
# Initial value for the number of R3load jobs for group Large. The value
# is also the minimum to which the number of jobs will be decreased.
initialLargeJobNum=5
# Initial value for the number of R3load jobs for group Small. The value
# is also the minimum to which the number of jobs will be decreased.
initialSmallJobNum=5
# If set to true migmonctrl stops to do changes to the order_by.txt.
pause=false
# Number of R3load jobs used for group Rowstore.
jobNumRowstore=33
# If set to true Rowstore tables are put in a separate group named
# Rowstore. R3load jobs for this group are started after the last table
# specific package has finished.
loadRowstoreSeparate=true
```

You can change the parameters also while *Migration Monitor* is running, because they are read every 5 seconds by default.

To check the add-on version, run the following command:

```
./sapjvm/sapjvm_6/bin/java -jar migmonctrl.jar -V
```

Log files of the add-on are `MIGMONCTRL.LOG` and `MIGMONJOBBER.CONSOLE.LOG`.

### 3.3.5 Using an SAP HANA Standby Server for the Import of SAP BW

If scale-out is released for your product, consider to use an SAP HANA standby server node for the import to improve performance. For this, consider the following:

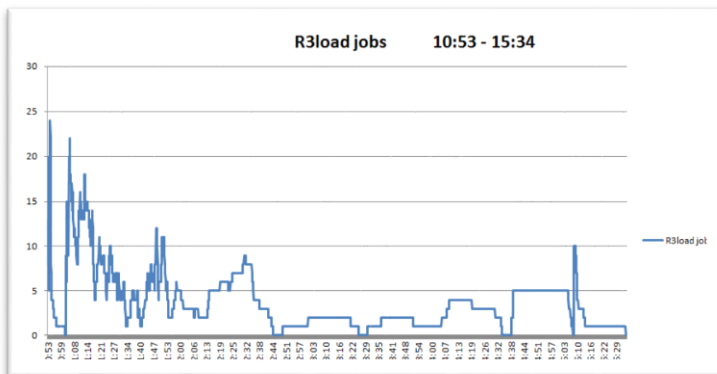
1. Perform a *Distributed System* installation.
2. Install the ASCS instance on the SAP system host.
3. Perform the Database Instance installation (including data import) on the SAP HANA standby node.
  - a. As a prerequisite, copy the profiles directory `/sapmnt/<SAPSID>/profile` from the SAP server, create user `<sapsid>adm` and group `sapsys` with the same IDs as on the ASCS instance host and change the owner of the copied profile directory to `<sapsid>adm:sapsys`.
  - b. You can use default settings (*Typical Mode*), but should consider to enable parallel export and import, if released for your product (see above). *Migmon* does not have to be started manually.
  - c. Enter the path to the newest kernel from SAP Service Marketplace.
4. Install the Primary Application Server Instance on the SAP system host.

### 3.3.6 Analysis of Import

For further analysis after the import and optimization of further import runs, you can use the files `R3LOAD_JOBS.CSV`, `FREE_MEMORY.CSV`, `SAVEPOINT_STATISTIC.CSV` and `CPU_USAGE.CSV`. The easiest way to use these files is to import them into Microsoft Excel, for example:

1. Open the CSV file in Microsoft Excel and mark all relevant values until the import has stopped (if you are in doubt, check the import log file for the exact time).
2. Select *Insert* → *Line* → *Line*.
3. In the inserted diagram, right-click on the values from the horizontal (category) axis and select *Format Axis....*
4. In the shown *Axis Options* dialog, select the value *Text axis* for the parameter *Axis Type*.

Result would be a diagram like the one taken from the example later in this documentation:



## 4 CONSIDERATIONS AFTER THE MIGRATION

### 4.1 Consistency Checks

To verify that the import has been performed successfully, *software provisioning manager* performs *Package Checker* and *Object Checker* runs as part of the default procedure. In addition, you can manually perform *Table Checker* – for more information, see SAP Note [784118](#).

- The *Package Checker* is used to verify that the import of all packages was at least started.
- The *Object Checker* is used to verify that all objects (tables, views, indexes, primary keys) are successfully created/loaded in database.
- The *Table Checker* is used to verify if the number of rows been exported is the same as the number of rows in the database (not usable in Socket mode).

To run *Table Checker* manually:

- Download `MIGCHECK_<version>.SAR` from SAP Service Marketplace at:  
<http://support.sap.com/swdc> → *Support Packages and Patches* → *Browse Our Download Catalog* → *Additional Components* → *SYSTEM COPY TOOLS GEN* → *SYSTEM COPY TOOLS GEN* <release> → # OS independent.
- Proceed as described in the Migration Checker Users' Guide that is part of the archive downloaded above.

### 4.2 Post-Processing

After migrations to **SAP BW on SAP HANA only**:

- Before performing other post-processing activities, run the ABAP report `RS_BW_POST_MIGRATION` as documented in the system copy guide.
- Run the ABAP report `RSDU_TABLE_CONSISTENCY` to identify potential problems, such as with database indexes, table partitioning, table distribution, and table classification – for more information, see [SAP Note 1937062](#) (Usage of `RSDU_TABLE_CONSISTENCY`).  
Before you run the report, make sure that you applied all SAP Notes listed in the attachment `REQUIRED_CORRECTION_NOTES.TXT` of SAP Note [1908075](#).
- SAP Note [1695112](#) (*Activities in BW after migrating to the SAP HANA database*) provides additional information on post-processing steps.
- See SAP Note [1949273](#) (*Important SAP Notes for SAP BW 7.40, powered by SAP HANA*) resp. SAP Note [1846493](#) (*Important SAP Notes for SAP BW 7.3x, powered by HANA*) prior to the production use of your SAP BW system migrated to SAP HANA

Apart from this, no additional post-processing activities are known as of now besides the standard activities – for more information, see the *System Copy Guide*.

### 4.3 Troubleshooting

#### 4.3.1 General Information

See the blog [Migration to SAP HANA: Analyzing Problems within Software Provisioning Manager](#) to get hints how to identify the root cause of issues during the classical migration procedure.

#### 4.3.2 Increase Trace Level of R3\* Tools

If R3load or any other R3\* tool stops, it may be useful to increase trace level and repeat the failing step. For this, set the following environment variable and re-run the tool:

- R3load: `R3LOAD_TL = 3`
- R3ldctl: `R3LDCTL_TL = 3`
- R3szchck: `R3SZCHK_TL = 3`



#### 4.3.3 Increase Trace Level of SAP HANA

Consider to increase the trace level for the SAP HANA SQLDBC client by using the following commands:

Action	Command
<b>Enable trace*</b>	<code>./hdbclient/hdbsqldbc_cons trace sql on</code>
<b>Refresh trace*</b>	<code>./hdbclient/hdbsqldbc_cons trace refresh</code>
<b>Disable trace</b>	<code>./hdbclient/hdbsqldbc_cons trace sql off</code>
<b>Locate trace</b>	<code>./hdbclient/hdbsqldbc_cons show config</code>

\*) While “*enable trace*” enables the tracing only for newly started processes, “*refresh trace*” enables the tracing also for already running processes.

#### 4.3.4 Checking Troubleshooting Information in SAP Notes

In addition to the SAP Notes for the migration procedure listed in previous sections, here is a selection of further SAP Notes containing information for specific issues:

- SAP Note [1722395](#) (*SAP HANA: Known problems*) lists known issues that may occur during a migration to SAP HANA
- SAP Note [1860493](#) (*Out of memory during import migration to HANA*) in case you are facing an out-of-memory error during the migration to SAP HANA
- SAP Note [1641210](#) (*SAP HANA database: Checking for suspected problems*) provides a checklist how to investigate a potential problem in the SAP HANA database

#### 4.3.5 Open an SAP Support Message

See SAP Note [752505](#) (*SAP System Provisioning / Installation Problems*). It describes the log files and which information is required in an SAP incident message so that it can be efficiently processed.

With this information, create an SAP incident with a direct connection to the customer system under the following component:

- In case of migration tool issues: BC-INS-MIG-TLA
  - Also see SAP Note [1797140](#) for a list of recommended steps to be performed before creating a message in this component.
- In case of general migration procedure issues: BC-INS-MIG
- In case of database-related issues: BC-DB-HDB-INS

## 5 APPENDIX

### 5.1 Example – Optimizing the Runtime for Migration of SAP BW Test System

In this section, you get an example how the runtime of an exemplary test run of a migration of an SAP BW 7.30 system (size in source database: 560 GB) to SAP HANA could be optimized. Although the runtimes and numbers come from a real project, other optimizations might be reasonable depending on your environment (such as used hardware).

In this example, the following optimizations were used:

- Two runs were performed – first run with standard parameters to identify optimization potential, second run with optimized parameterization
- Export and import ran in parallel
- Table splitting was used
- Import was done on an additional SAP HANA standby node to profit from the strong hardware
- *Massloader* option was used in the second run to accelerate the import

As a scale-out landscape was used, SAP HANA landscape redistribution was performed automatically to prevent import errors and ensure a properly working system. For this, SAP HANA landscape redistribution optimally distributes database tables over all database nodes in the scale-out landscape.

#### 5.1.1 First Run

##### *Table Splitting Preparation:*

The procedure was executed to split the biggest tables in the database:

- <TABLE\_01> (798.199.689 rows, 110 GB): 11 splits
- <TABLE\_02> (216.663.239 rows): 5 splits

For table <TABLE\_01>, the run of *R3ta* in the first migration took about 6 hours.

##### *Database Instance Export:*

- Export was configured to run in parallel mode, which means special signal files (SGN files) were created in a special exchange directory on the export medium to signalize the import side that packages have finished and their import can start.
- Mostly, default *package splitter* settings were taken (“extract 10 biggest tables from each package”), maximum size of all tables inside a package was changed to 300 MB.
- *order\_by.txt* file was used, packages were sorted by name and two packages (<TABLE\_03>\_1 and <TABLE\_03>\_2) were moved to the beginning, as longer runtime was expected for them (due to the high number of tables they comprised and based on experience made in other projects):  
    <TABLE\_03>\_1 (9869 tables)  
    <TABLE\_03>\_2 (3631 tables)  
    <Alphabetical order of rest of tables>

##### *Target System (SAP HANA) Setup and Import:*

1. ASCS instance was installed on the SAP system host.
2. Database instance was installed (including the actual migration) on the SAP HANA standby node (see detailed description above).
  - a. As a prerequisite, the profiles directory `/sapmnt/<SAPSID>/profile` was copied from the SAP server and the owner was set to `<sapsid>adm:sapsys`, which was also created upfront.
  - b. Parallel export and import was chosen.
  - c. `721_EXT SAPEXE_DB.SAR` and `SAPEXE.SAR` were configured to be installed in addition to the `720_EXT` kernel.
  - d. SAP HANA landscape reorg was executed (as outlined above).
3. Primary Application Server Instance was installed on the SAP system host.

*Migration Times:*

- Export: 3h 55min (10:46 – 14:41)
- Import: 4h 41min, 4h 31min data import
  - Table creation: 10:53 – 11:01
  - SAP HANA landscape reorg + data load: 11:03 – 15:34
- **Complete runtime: 4h 48min**

## Longest running packages on the export side:

package	time	start date	end date
<TABLE_01>-11	2:43:22	<DATE> 11:58	<DATE> 14:41
<TABLE_01>-5	2:29:43	<DATE> 12:09	<DATE> 14:38
<TABLE_01>-6	2:25:42	<DATE> 12:13	<DATE> 14:39
<TABLE_01>-7	2:24:52	<DATE> 12:14	<DATE> 14:39
<TABLE_01>-8	2:24:49	<DATE> 12:14	<DATE> 14:39
<TABLE_01>-9	2:19:43	<DATE> 12:19	<DATE> 14:39
<TABLE_01>-2	2:07:01	<DATE> 11:58	<DATE> 14:05
<TABLE_01>-3	2:06:02	<DATE> 12:01	<DATE> 14:07
<TABLE_01>-4	2:01:28	<DATE> 12:07	<DATE> 14:09
<TABLE_04>	1:47:17	<DATE> 10:46	<DATE> 12:34
<TABLE_05>	1:47:09	<DATE> 11:49	<DATE> 13:36
<TABLE_06>	1:36:10	<DATE> 11:13	<DATE> 12:49
<TABLE_01>-10	1:32:06	<DATE> 11:57	<DATE> 13:29
<TABLE_01>-1	1:04:44	<DATE> 11:56	<DATE> 13:01
<TABLE_07>	0:58:11	<DATE> 10:50	<DATE> 11:48
<TABLE_08>	0:49:15	<DATE> 11:24	<DATE> 12:14
<TABLE_09>	0:43:09	<DATE> 11:30	<DATE> 12:13

...

Package <TABLE\_01>-11 was started at 11:58 and ran until 14:41, which was also the end of the export. On the import side, keep in mind that column-store tables are loaded first. After that, row store is loaded.

## Longest running packages on the import side:

## Column store

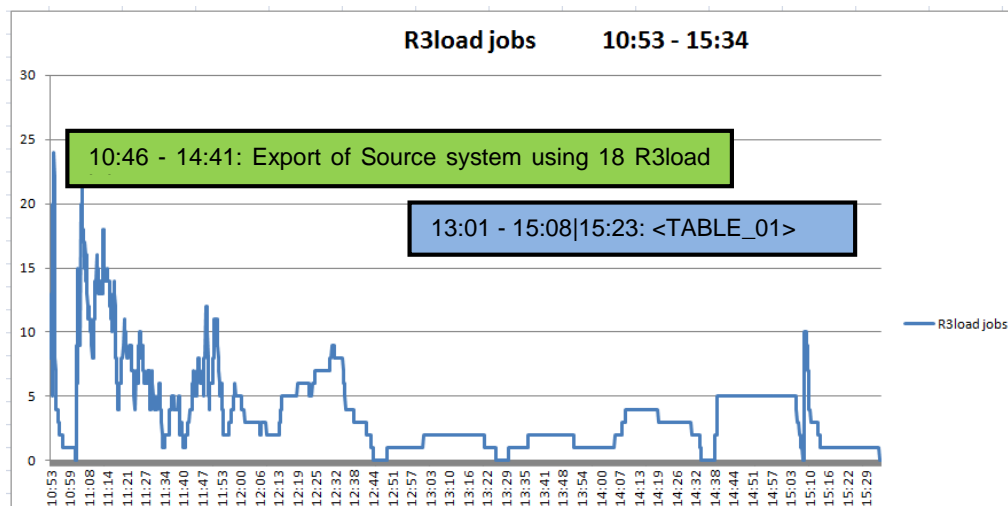
package	time	start date	end date
<TABLE_03>_1__T	0:47:10	<DATE> 11:07	<DATE> 11:55
<TABLE_05>__T	0:43:26	<DATE> 13:36	<DATE> 14:20
<TABLE_06>__T	0:32:26	<DATE> 12:49	<DATE> 13:22
<TABLE_01>-5__TPIMU	0:28:37	<DATE> 14:39	<DATE> 15:08
<TABLE_01>-9__TPIMU	0:28:35	<DATE> 14:39	<DATE> 15:08
<TABLE_01>-6__TPIMU	0:27:22	<DATE> 14:39	<DATE> 15:07
...			
<TABLE_01>__DTP	0:14:55	<DATE> 15:08	<DATE> 15:23

The column store has finished at 15:08 with the data import of table <TABLE\_01>. The index creation, merge and compress were finished at 15:23. As the migmonctrl.jar add-on of *Migration Monitor* does not take notice of *R3load* post jobs, it started the import of the row-store tables already at 15:08.

## Row store

package	time	start date	end date
<TABLE_04>__T	0:20:59	<DATE> 15:08	<DATE> 15:29
...			
SAPVIEW__T	0:04:06	<DATE> 15:29	<DATE> 15:34

The row store has finished at 15:34 with a runtime of 26 minutes.



Import into SAP HANA:

- 340 jobs in total
- 10:53 – 11:03: Creation of tables and execution of SAP HANA landscape reorg
- 11:03 - 15:34: Import of data
- 15:08: Start of import of row-store tables

### 5.1.2 Second Run

*Changes and Results:*

Based on the learnings from the first run, the following changes were performed for the second run:

- For the second run, we used `R3ta_hints.txt (<TABLE_01> RECORDS)` for table splitting preparation to decrease the runtime of `R3ta` from about six hours to about one hour.
- `order_by.txt` was adapted as follows:
  - Moved table `<TABLE_01>`, `<TABLE_02>`, and `<TABLE_04>` upwards, so that they get exported earlier
  - Split `<TABLE_04>` in two pieces
  - Adjusted *Package Splitter* to extract the biggest 50 tables
  - Adjusted *Package Splitter* to use a maximum size of a package of 150MB before splitting the package
- Enabled *massloader* in *R3load* by setting the environment variable `HDB_MASSLOADER=YES` before starting *SAPinst*

*Migration Times:*

- Export: 3h 25min (10:12 – 13:37)
- Import: 4h 11min, 4h data import
  - Table creation: 10:15 – 10:23
  - SAP HANA landscape reorg + data load: 10:26 – 14:26
- **Complete runtime: 4h 14min**

Longest running packages on the export side:

package	time	start date	end date
<TABLE_01>-2	2:33:48	<DATE> 11:01	<DATE> 13:35
<TABLE_01>-4	2:31:32	<DATE> 11:04	<DATE> 13:36
<TABLE_01>-3	2:31:17	<DATE> 11:03	<DATE> 13:35
<TABLE_01>-5	2:30:55	<DATE> 11:05	<DATE> 13:36
<TABLE_01>-6	2:29:32	<DATE> 11:06	<DATE> 13:36
<TABLE_01>-7	2:28:13	<DATE> 11:08	<DATE> 13:36
<TABLE_01>-8	2:24:17	<DATE> 11:12	<DATE> 13:36
<TABLE_01>-9	2:13:20	<DATE> 11:23	<DATE> 13:37
<TABLE_01>-10	2:01:20	<DATE> 10:55	<DATE> 12:56
<TABLE_01>-1	1:12:16	<DATE> 10:53	<DATE> 12:05
<TABLE_05>-2	0:58:49	<DATE> 11:28	<DATE> 12:27
<TABLE_05>-1	0:55:47	<DATE> 11:25	<DATE> 12:21
<TABLE_04>-2	0:46:51	<DATE> 10:49	<DATE> 11:36

...

Package <TABLE\_01>-9 was started at 11:23 and ran until 13:37, which was the end of the export.

Longest running packages on the import side:

Column store

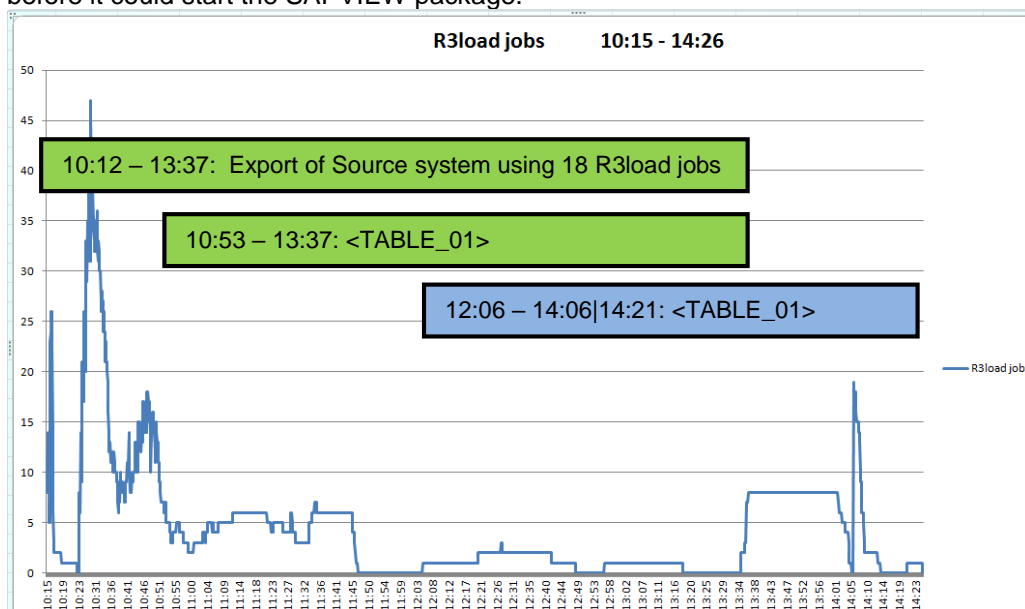
package	time	start date	end date
<TABLE_03> 1 T	0:50:13	<DATE> 10:32	<DATE> 11:22
<TABLE_01>-9 TPIMU	0:28:58	<DATE> 13:37	<DATE> 14:06
<TABLE_01>-4 TPIMU	0:28:49	<DATE> 13:37	<DATE> 14:05
<TABLE_01>-6 TPIMU	0:28:40	<DATE> 13:37	<DATE> 14:05
<TABLE_01>-7 TPIMU	0:28:38	<DATE> 13:37	<DATE> 14:05
<TABLE_01>-8 TPIMU	0:27:35	<DATE> 13:37	<DATE> 14:04
<TABLE_01>-3 TPIMU	0:27:23	<DATE> 13:35	<DATE> 14:02
<TABLE_01>-5 TPIMU	0:27:11	<DATE> 13:36	<DATE> 14:03
...			
<TABLE_01>__DTP	0:15:12	<DATE> 14:06	<DATE> 14:21

The column store finished at 14:21 (the import of <TABLE\_01> finished at 14:06, but the merge and the optimization compression took another 15 minutes). As the migmonctrl.jar add-on of *Migration Monitor* does not take notice about *R3load* post jobs, it started the import of row-store tables already at 14:07.

Row store

package	time	start date	end date
<TABE_10>__T	0:07:34	<DATE> 14:07	<DATE> 14:14
<TABLE_11>__T	0:06:42	<DATE> 14:07	<DATE> 14:13
<TABLE_04>-2 TPIMU	0:02:14	<DATE> 14:07	<DATE> 14:09
...			
SAPVIEW__T	0:04:17	<DATE> 14:22	<DATE> 14:26

Import of row store tables finished at 14:14, which means it only took 7 minutes due to the *massloader* option. After that, *Migmon* waited for 8 minutes for the last column store table to be merged and compressed, before it could start the SAPVIEW package.



Import into SAP HANA:

- 481 jobs in total (more jobs than in first run due to increased number of splits)
- 10:15 – 10:23: Creation of tables and execution of SAP HANA landscape reorg
- 10:26 – 14:26: Import of data
- 14:07: Start of import of row-store tables

### **5.1.3 Potential for Further Optimization**

If the table <TABLE\_01> would have been exported right from the beginning with its runtime of 2h 44min, the process would have finished already at 12:56. With the overlap of 40 minutes, it could have been imported at 13:40. Including row store tables, this would have resulted in an end time of 13:50 and in a runtime of 3h 38min (10:12 – 13:50), still with the potential to decrease the time even more by raising the amount of splits.