

# SAP on IBM i

## SAP Software Update Manager: Behaviors and Insights



This document can be found on the web at [www.ibm.com/support/techdocs](http://www.ibm.com/support/techdocs)

Version: 1.0 August 2018



## Table of Contents

Table of Contents .....	2
1 Preface.....	4
1.1 Abstract .....	4
1.2 Applicability.....	4
1.3 Author Bio.....	5
1.4 Special Thanks to .....	5
1.5 Disclaimer .....	5
1.6 Document Change History .....	5
2 Introduction.....	6
3 SAP Software Update Manager .....	7
3.1 Overview .....	7
3.2 Prerequisites .....	9
3.2.1 SAP Host Agent .....	9
3.2.2 Minimum SAP Kernel Release Level For Use With SUM .....	10
3.2.3 SAP User Concept.....	11
3.2.4 SAP JVM.....	11
3.2.5 Kernel Replication .....	11
3.3 SUM Components .....	12
3.4 Important File System Parts .....	14
3.4.1 Top-level Subdirectory Structure .....	14
3.4.2 Software for ABAP Upgrades .....	14
3.4.3 Software for Java Upgrades .....	14
3.4.4 SUM Specific Java Runtime .....	14
3.4.5 Cryptographic Software .....	15
3.4.6 Remaining Parts .....	15
3.5 SUM running on IBM i Machines.....	16
3.5.1 Minimum OS Release .....	16
3.5.2 Identifying SUM Processes.....	16
3.5.3 Independent Auxiliary Storage Pool .....	16
4 Software Update Manager Processing in Detail .....	18
4.1 Starting and Stopping SUM .....	18
4.1.1 Web Browser Access.....	18
4.1.2 Running Multiple SUMs in Parallel.....	18
4.2 Using the SUM GUI.....	19
4.2.1 GUI Options .....	19
4.2.2 Roadmap Sections .....	23
4.2.3 User Defined Breakpoints .....	26
4.3 Upgrade Shadow Instances .....	26
4.3.1 Shadow Database Libraries .....	26
4.3.2 Shadow Instance File System.....	32
4.4 Important Phases for Technical Decisions.....	32
4.4.1 ABAP Upgrade Phase INITSUBST .....	32
4.4.2 ABAP Upgrade Phase ASCS_ASK.....	35
4.5 Important Platform Specific Phases .....	35
4.5.1 Platform Specific Phases for ABAP Stack Upgrades.....	35
4.5.2 Platform Specific Phases for Java Stack Upgrades .....	37

5	Performance Considerations .....	39
5.1	Long Running Upgrade Phases .....	39
5.1.1	ABAP Phase EHP_INCLUSION .....	40
5.1.2	ABAP Phase CONFLICT_CHECK .....	41
5.1.3	ABAP Phase DDIC_UPG .....	41
5.1.4	ABAP Phase SHADOW_IMPORT_INC .....	42
5.1.5	ABAP Phase TABIM_UPG .....	42
5.1.6	Other Relevant Phases .....	42
5.1.7	Long Running Phases in Java Upgrades .....	43
5.2	Parallelism During the Upgrade .....	43
5.2.1	Upgrade Settings .....	43
5.2.2	Hardware Considerations .....	44
5.2.3	Parallel Considerations for Programs tp and R3trans .....	45
5.2.4	Archiving and Reorganizing Database Tables .....	45
5.2.5	Parallelism for Modification Imports .....	47
5.3	SUM Performance Checklist .....	48
6	Special Upgrade Strategies .....	50
6.1	Combined Upgrade and Unicode Conversion .....	50
6.2	Near Zero Downtime Maintenance .....	52
6.3	Zero Downtime Option .....	54
6.4	Upgrade with Database Migration Option .....	54
7	Conclusion .....	55
8	Table of Figures .....	56
9	References .....	57
9.1	IBM Websites .....	57
9.1.1	IBM Support and Knowledge Center .....	57
9.1.2	IBM Techdocs .....	57
9.2	SAP Websites .....	57
9.2.1	SAP Community and Help Portal .....	57
9.2.2	SAP Wiki and Blogs .....	58
9.3	Other Websites .....	58
10	Abbreviations .....	59
11	Trademarks .....	60

# 1 Preface

## 1.1 Abstract

IT systems running SAP business software are usually critical systems and important to run business processes successfully. Keeping such IT systems at a current level ensures the correctness and reliability of the business decisions based on data of those systems. Therefore, the software life cycle management is an essential part of SAP system maintenance in your IT landscape.

This includes software corrections to improve performance or to eliminate minor bugs. In parallel, and independent of the main SAP software development, legal changes may occur. Updated software components will implement conditions and regulations if legislation has changed. Finally, further improvements and new features will be provided by the software development to enrich the value of the SAP business software.

To reach the goals mentioned above, it is sometimes necessary to take an SAP system off-line for a certain amount of time to run the required maintenance tasks. This off-line time is very critical since the affected SAP system cannot be used during that time. The goal must be to keep this time frame as short as possible.

Over the past years, management of software life cycle tasks have evolved to the point that SAP has consolidated and bundled different tools and invented the **SAP Software Update Manager (SUM)**. This tool provides different ways to run maintenance tasks and drives the user step by step through the process using a modern and unique browser-based GUI.

The joint SAP and IBM development team works continuously to improve the interoperability between SAP software components with the latest technologies and database features of the IBM i platform. This document describes platform specific details of the SAP upgrade process and is a complete rework of a formerly released white paper that was last published in January 2012 on the IBM Techdocs web page.

## 1.2 Applicability

This white paper applies to any maintenance operation of SAP systems based on SAP NetWeaver using the tool SAP Software Update Manager running on the IBM i platform. It is valid for all SAP system stack types (such as ABAP, Java or Dual stack systems) and covers the SUM versions 1.0 and 2.0. The upgrade tool SUM is used for following software maintenance scenarios:

- implementing an SAP Support Package stack within a given SAP product,
- applying an SAP Enhancement Package stack for an SAP product or
- upgrading to a higher release of a given SAP product

The white paper explains technical aspects when running the SUM tool on an IBM i server. It focuses on the configuration and performance aspects of the upgrade process, the special behavior of the upgrade tools when running on the IBM i platform, and explains improvements made for that platform. Insights about the construction of the so-called “shadow instances” and their databases are included, as well as descriptions of upgrade phases which run specific to this platform.

### 1.3 Author Bio

Author: René Lang, Company: IBM Germany, Research and Development Lab

René Lang is working for the IBM Germany, Research and Development Lab, as a member of the *SAP on IBM i Development* team located in St. Leon-Rot, Germany. He started his carrier at an IBM subsidiary company in 1993 working in different projects and is developing customer specific applications running on IBM i. In February 2002, he joined the SAP on IBM i Development team as IT Specialist focusing on platform specific adoption and further enablement of the SAP upgrade tools. The author can be reached via email at [rene.lang@de.ibm.com](mailto:rene.lang@de.ibm.com).

### 1.4 Special Thanks to

- Dr. Thomas Obermeier, SAP SE
- Christian Bartels, IBM Germany, Research and Development Lab
- Christine Gaul-Gaensslen, IBM Germany, Research and Development Lab
- Ron Schmerbauch, IBM US, Rochester Development Lab
- Dr. Stephan Kammerer, IBM Germany, Research and Development Lab
- Christian Hölters, SAP SE

### 1.5 Disclaimer

Driving maintenance tasks for your SAP production system are critical tasks that require careful planning and execution. This white paper is not an upgrade guide and not intended to replace it. There are some points of osculation between this white paper and the SAP documentation for better understanding. Please refer to the latest official SAP Upgrade Guides available in the SAP Help Portal (<https://help.sap.com/>) for detailed documentation about the upgrade process. For some SAP information referenced by this document (e.g. links to SAP Notes), a valid and registered SAP support user profile (a so-called “S-user”) is required. If access to a restricted content section is needed, please contact a user administrator in your company. If required, an [SAP Local Support Center](#) can assist you in identifying one of them.

### 1.6 Document Change History

Document created on: June 2018

Last update on: 10 August 2018

Version	Date	Changes
1.0	August 2018	Document first release

## 2 Introduction

Driving any kind of business is a challenging task. Lots of conditions and relations need to be managed, business models must be developed and adopted over time. Since nowadays almost everything is interconnected with the world-wide web, the time slots for appropriate reactions get shorter and the competition in the market get more intensive. The question is not, if you want to use software to support your business closely, but what kind of software to use. One of the most well-known vendors of business software is SAP SE.

SAP provides a wide variety of software products. Besides new products acquired from other companies like SAP BusinessObjects® or SAP Concur®, SAP continuously develops and improves products based on its own SAP NetWeaver® technology, such as products from the SAP Business Suite® or SAP's core enterprise resource planning software SAP ECC®. This document describes the behaviors and insights of the maintenance process using the SAP Software Update Manager on platform IBM i for SAP products based on SAP NetWeaver.

Upgrading already installed SAP software requires consideration of several constraints. You should plan an appropriate maintenance window for the time needed to execute the upgrade. You should also get familiar with the corresponding user guides and requirements that need to be fulfilled for hardware and software to continue running your business software after the upgrade. The required software for the upgrade needs to be ordered or downloaded. A downtime window needs to be chosen when the SAP software can be unavailable or running in a restricted mode without full functionality.

The upgrade of an SAP system can be executed in 3 different scenarios. All scenarios are executed using the SAP Software Update Manager:

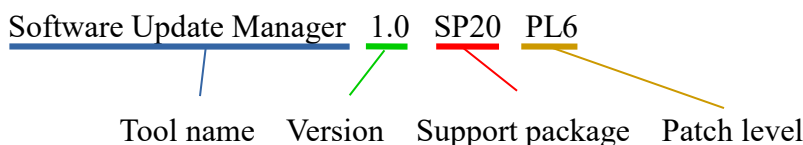
- Applying a Support Package stack to an SAP system allows you to get recent fixes for already installed SAP components. This often includes components specific for the underlying database or operating system. The SAP release itself will remain the same.
- With an Enhancement Package stack, the SAP system can get new features or business functions for a given SAP product release. It is less complex to be applied compared to a release upgrade but more complex than applying a Support Package stack. Each Enhancement Package provides its own Support Package line.
- Running a release upgrade is the most complex maintenance scenario. This raises the SAP system to a higher SAP product release, which provides completely new application capabilities that are adopted to newer database and operating system versions.

## 3 SAP Software Update Manager

### 3.1 Overview

In the past, SAP has provided a set of different tools to maintain existing SAP systems. To apply support packages to an ABAP based SAP system, transaction `SPAM` (Support Package Manager) was used, and for Java based systems the Java tool `JSPM` (Java Support Package Manager). The first enhancement packages were installed with tool `SAPehpi` (SAP Enhancement Package Installer), and for release upgrades, specific versions of tool `SAPup` (SAP Upgrade for ABAP systems) or `SAPJup` (SAP Java Upgrade) have been developed, depending on the respective SAP target product. In some cases, special maintenance tools for certain SAP products were provided, such as the `SolManUp` for SAP Solution Manager systems or the `CEUpdateManager` for SAP Composition Environment systems.

The SAP Software Update Manager is an evolution and consolidation of all the former upgrade and maintenance tools and has been SAP's multi-purpose tool to run any maintenance operation for active SAP systems since 2011. The development for SUM follows an iterative and incremental principle. With that, SAP delivers enhanced and corrected SUM packages on a quarterly basis as part of the SAP Software Logistics Toolset. These software packages are counted as Support Package stacks. Each part of a Support Package stack counts its own patch level, which gets increased for shipments between two support package stacks, e.g. for bug fixing. This means, to address a certain SUM version, the following notation is used:



The Software Update Manager is available in two versions. In September 2017, SUM version 2.0 was shipped for the first time. It will be maintained in parallel to version 1.0, which has been available for several years. Fixed rules determine which version is used as follows, with a few exceptions for special upgrade scenarios:

- SUM version 1.0 for SAP target releases below 7.50 (for all stack types) and for Java stacks based on 7.50 or higher.
- SUM version 2.0 for SAP target releases based on 7.50 or higher and for ABAP stacks only.

Starting with SUM 1.0 SP15 in 2012, the tool provides a new and modern user interface (UI). This UI is browser-based and uses the Representational State Transfer (REST) programming paradigm. It enables stateless communication between the SUM tool and the user. The SUM can be controlled from any browser in the local network, since it provides the needed web server by itself. After the initial registration in the SAP Host Agent, SUM can be started by calling a web address in the browser from any computer in that network (see [figure 1](#) on page 8).

Not all upgrade paths are technically possible in a single step. For example, a minimum of two steps are required to upgrade all the way from start release SAP R/3 4.7 (based on SAP\_BASIS 6.40) to SAP ECC 6.0 EHP8 (which is based on SAP\_BASIS 7.50). A detailed overview of supported upgrade paths can be found at SAP's [Product Availability Matrix](#) (PAM).

One of the most powerful features of SUM compared to its predecessors like `SAPup` or `SAPJup` is the capability to work with any SAP system and SAP release. This flexibility requires to have additional descriptive or functional software components on-site. This set of software consists of:



- The SAP system to be maintained of any application server type, either ABAP, Java or dual-stack systems containing both application server types.
- An SAP Solution Manager system that knows the current configuration of the SAP system to be upgraded in order to calculate and propose all possible maintenance options. Based on this calculation it provides a list of software archives or media images to be downloaded from SAP Software Download Center. This list of software is also called the upgrade stack. The upgrade stack also contains the most recent version of the SUM tool, but it can be deselected and replaced by a specific version, e.g. when needed for a larger project. The Solution Manager system may run on any host and is only needed to make the mentioned calculation before a SUM maintenance.
- The SAP Solution Manager also generates a Stack XML file, which contains the current and the planned target configuration of the given SAP system. SUM needs this information to prepare and execute the system maintenance operation. Therefore, the file needs to be correlated with the downloaded upgrade stack. At the beginning of processing, SUM verifies if the Stack XML and the upgrade stack fit together and are consistent.
- A well configured and up-to-date SAP Host Agent must run on the same host as the SAP system. The SAP Host Agent provides different services for SUM, most importantly the launch of the technically needed processes.

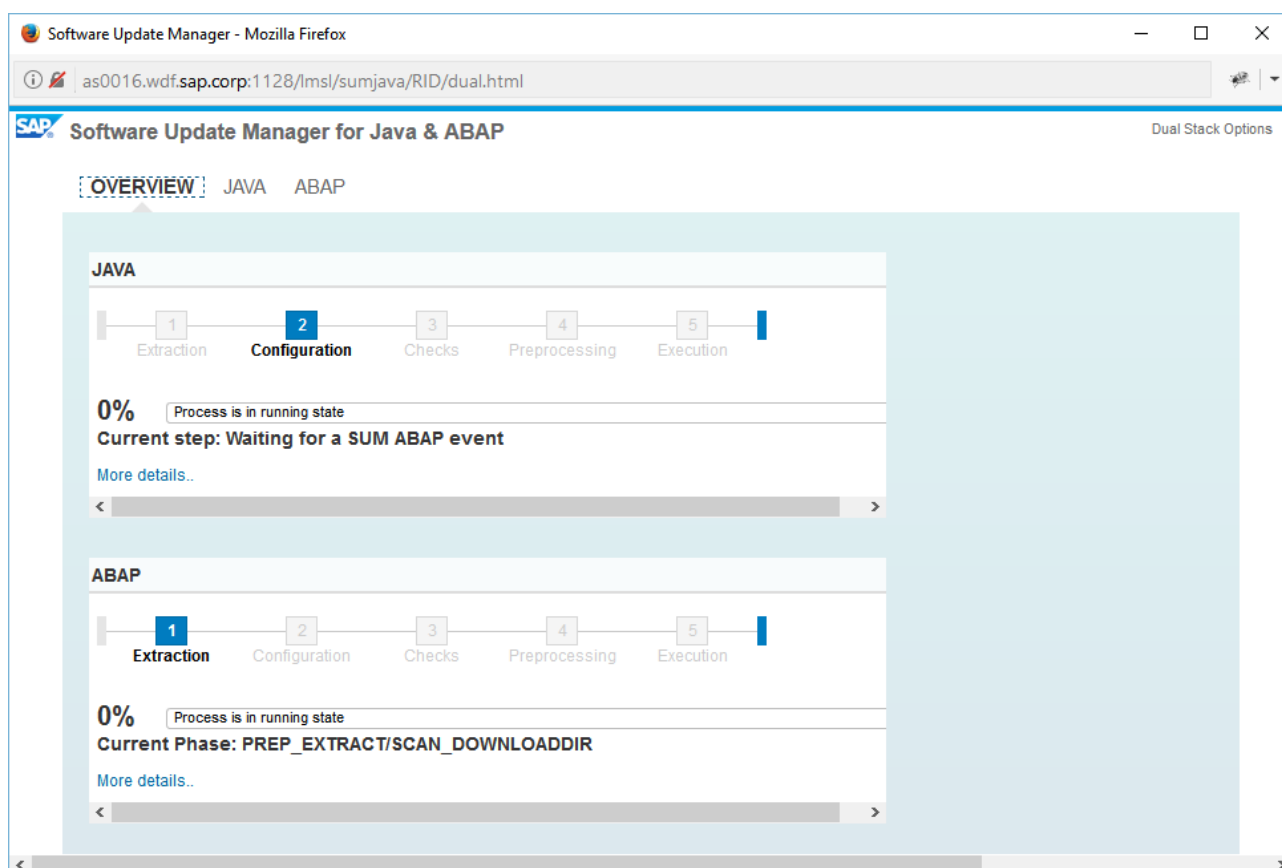


Figure 1: SUM 1.0 for a dual-stack SAP system at the beginning

For every SUM Support Package level, SAP provides a large set of documentation. First source of information is the corresponding Upgrade Guides that can be found in the [SAP Help Portal](#). Besides



general guides with information about planning for system maintenance, there are different guides available for each platform and application server configuration type. Choose your guides based on the planned target system configuration. The easiest way to find the right guide is using the [SAP NetWeaver Guide Finder](#).

To describe the latest changes and corrections for a given SUM version, there are SAP Notes released specific to the SUM SP level as well. These SAP Notes always appear in pairs – one platform independent and one platform specific SAP Note. The platform independent SAP Note describes the feature scope of the SUM version and points to additional SAP Notes for supported upgrade target releases. Finally, it provides information about known issues in specific upgrade phases that may appear independent from a given DB/OS combination. The documentation is valid for all patch levels of a SUM version. Maybe additional SAP Notes need to be considered for certain patch levels. Since the SAP Notes will be maintained as long as the respective SUM version is in support, it is necessary to ensure having the latest version of the SAP Note in place. For the current versions of SUM<sup>1</sup>, the following SAP Notes must be used for platform IBM i:

- [SAP Note 2472928](#) Central Note – Software Update Manager 1.0 SP22 [lmt\_004]
- [SAP Note 2536658](#) IBM i: Additional Information – Software Update Manager 1.0 SP22
- [SAP Note 2529257](#) Central Note – Software Update Manager 2.0 SP02 [lmt\_022]
- [SAP Note 2593059](#) IBM i: Additional Information – Software Update Manager 2.0 SP02

The SAP SL Toolset 1.0 SP23 consists of SUM 1.0 SP22, SUM 2.0 SP02 and other software maintenance tools such as SAP Software Provisioning Manager (SWPM), SAP ABAP Support Package Manager (SPAM) and more. Meanwhile, the SP level of the whole toolset is counted different compared to its components due to different shipment strategies.

## 3.2 Prerequisites

There are a few prerequisites to get SUM running on your IBM i server which typically need to be fulfilled before the upgrade starts. Some of them may require additional downtime, if the affected SAP system is not yet at the needed level. SUM checks the requirements and will stop if they are not met. For the ABAP part of SUM, this is done in the upgrade phase `PREP_INFRASTRUCTURE_OS4`, and for the Java part it is done in phase `OS4-PREREQ-CHECKS` at the very beginning of the upgrade procedure. All subsequently referenced patch levels are subject to change and may be increased with future SUM versions. Prerequisite checks allow SUM to avoid known and already fixed issues.

### 3.2.1 SAP Host Agent

Since all current versions of SUM get started via browser request to the SAP Host Agent, there is a need to have a minimum patch level of the host agent installed on your machine. This ensures the SUM starts correctly and with the right environment. The currently minimum required version is SAP Host Agent 7.21 patch level 25. As mentioned in the upgrade guide as well, it is recommended to use the latest version. If needed, a newer version can be installed without additional downtime of the SAP system. Descriptions for the installation of a new SAP Host Agent can be found in

- [SAP Note 1031096](#) Installing Package SAPHOSTAGENT

---

(1) Based on delivery of SAP SL Toolset 1.0 SP23 from May 2018

It is also important to configure the SAP Host Agent correctly for use with the SUM. This is done using the shell scripts `STARTUP` (for SUM 1.0) or `SUMSTART` (for SUM 2.0), called with parameter `confighostagent` as described in the SAP Upgrade Guide. For security reasons, the configuration of the SAP Host Agent needs to be done by user profile `QSECOFR`. User profile `<SID>ADM` does not have sufficient authority to do it, and SAP also does not recommend extending the special authorities of `<SID>ADM`, as it is technically not required. The default session settings for user profile `QSECOFR` might have been adopted for local country and language requirements, but these settings may be incompatible with EBCDIC codepage 500 or the US English language. In such cases, the job needs to be changed temporarily using the command `CHGJOB` as shown in [figure 2](#). After that modification, a shell session must be started using command:

```
CALL QP2TERM
```

To avoid any unwanted side effects, it is recommended to log off such a modified `QSECOFR` session after leaving the program `QP2TERM`. Command `QSH` also starts a shell session, but `QP2TERM` must be used as for any other SAP related shell script.

The screenshot shows a terminal window titled 'AS0016' with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main area displays the following text:

```

Command Entry
Request level: 4
AS0016
All previous commands and messages:
  Job 971144, QSECOFR/RENE1 started on 04/16/18 at 14:45:35 in subsystem
  QINTER in QSYS. Job entered system on 04/16/18 at 14:45:35.
  > /* */
  3 > CALL QCMD
  4 > CHGJOB LANGID(ENU) CCSID(500)

Type command, press Enter.
====> CALL QP2TERM

F3=Exit   F4=Prompt   F9=Retrieve   F10=Exclude detailed messages
F11=Display full   F12=Cancel   F13=Information Assistant   F24=More keys

MA  A  MW  18/019
|902 - Session successfully started

```

The text `QSECOFR` and `CHGJOB LANGID(ENU) CCSID(500)` are highlighted with red boxes in the original image. The prompt `====>` is followed by `CALL QP2TERM`, which is also highlighted with a red box.

Figure 2: Temporary session settings for user `QSECOFR`

### 3.2.2 Minimum SAP Kernel Release Level For Use With SUM

The SAP kernel is usually exchanged by the tool during the upgrade procedure. Nevertheless, some of the preparation steps are done in conjunction with programs of the source kernel release. Therefore, a minimum kernel release must be active to provide full support for SUM. In the present implementation, SUM will only continue if the SAP system uses at least kernel release 7.20 patch level 322. As the kernel release 7.20 is out of maintenance since a while, typically kernel releases 7.21 or 7.22 or

higher are used. The only exceptions are older SAP systems using kernel release 6.40 (e.g. SAP ECC 5.0) or kernel release 4.6D (e.g. SAP R/3 4.6C). All other kernel versions, such as releases 7.00, 7.01, 7.10 or 7.11, are out of maintenance and must be replaced by a currently supported 7.2x kernel version before using SUM. Applying a new SAP kernel requires a short maintenance downtime because the affected SAP instances need to be restarted to run with the new kernel.

### 3.2.3 SAP User Concept

Some years ago, the SAP on IBM i development team decided to adopt the general SAP user concept to the IBM i platform for compatibility reasons. In the past, several user profiles for different purposes specific to IBM i had been created at OS level. The new user concept was developed in order to reduce the number of user profiles significantly, using only two user profiles per SAP system – user <SID>ADM and group profile <SID>OWNER (where <SID> stands for the SAP system ID). The new user concept more accurately reflects the SAP way how an SAP system interacts with OS user profiles. On the platform IBM Db2 for i, there is still no dedicated user to manage the database. The database is permanently active on the IBM i platform and does not need to be installed separately as with other database management systems. Maintenance tasks related to the SAP database must be executed with user profile <SID>ADM as well. Besides this, the SAP application user profiles (both ABAP and/or Java users) are not related to this process and stay unchanged during the conversion. The switch of the user concept is a one-time, but time-consuming task, requiring a maintenance downtime, but this can be minimized by taking some actions described in the note that is mentioned below. After the switch, the old user profiles are kept at OS level, but they are deactivated. They can be safely deleted later after a review (e.g. check for owned objects or file system parts). To perform checks before the conversion of the user concept and convert the SAP system to the new user concept, you can use the tool CONVUSRCPT that is shipped with the SAP kernel and described in

- [SAP Note 1149318](#) User concept conversion using the tool CONVUSRCPT

### 3.2.4 SAP JVM

There are many Java virtual machines (JVM) and runtime engines from different vendors available for Java based products. To reduce external dependencies, SAP provides their own versions for their Java stack based products. If the SUM detects a Java stack in the SAP system, it verifies the usage of an SAP JVM. No vendor JVM, e.g. from IBM, is allowed any longer. For security reasons it is highly recommended to use always the latest available JVM patch level. SAP provides a separate tool based on the SAP Software Provisioning Manager (SWPM) to switch the JVM of an SAP system. A collection of questions and answers to this topic can be found in

- [SAP Note 1907153](#) SAP JVM Switch Tool: FAQ

### 3.2.5 Kernel Replication

Finally, but just as a recommendation, it makes sense to have automatic kernel replication and the new kernel directory structure available for the SAP system. These are not requirements for the SUM, but, once enabled, they make the kernel maintenance steps as requested by SUM easier (e.g. applying requested kernel patches). The automatic kernel replication is a mechanism where every SAP instance uses an own copy of the kernel executables (at least those executables needed to run that instance). The concept is based on a central file system directory, where the kernel master copy is stored. During

the start of an SAP instance, program `sapcpe` checks if there are newer versions of the kernel executables and copies them into an instance specific directory before the instance itself gets started. The kernel replication and details about the new kernel directory structure are described in the following SAP Notes:

- [SAP Note 1632754](#) IBM i: Changeover to instance-specific directory
- [SAP Note 1078134](#) IBM i: Distribution of ILE and PASE system components

### 3.3 SUM Components

The software archive for SUM version 1.0, which contains the upgrade tool, consists of two major parts: One part is used to upgrade ABAP stacks, the other one to upgrade Java stacks. In the past, SAP used different tools for such maintenance operations. When the Software Update Manager was introduced, SAP consolidated the different tools and merged them together to provide a single tool with a unified browser-based GUI frontend. For SUM version 2.0 (introduced in September 2017), the development had its focus on the support of ABAP stacks only

The ABAP upgrade tools are the biggest part of the SUM archive. It consists of many different components, such as C/C++ programs to run the upgrade procedure, including a small web server to enable the communication with a browser to control the whole process. These programs are usually programs for the Portable Application Solutions Environment (PASE), which is a special runtime environment derived from an AIX runtime. Other programs are delivered for the Integrated Language Environment (ILE). They are provided in a container (IBM i object type `*FILE`, attribute `SAVE`) and extracted automatically by the SUM procedure. Large file system parts of the ABAP component have the same structure as a common SAP transport directory. The SUM also delivers SAP transport files containing ABAP structures and reports needed for the upgrade. These ABAP parts are imported to the affected SAP system at the beginning of the upgrade. In addition, a lot of XML or plain text files are part of the archive, containing control values for certain upgrade steps, dialogs and templates that get filled step by step with the values determined by the specific upgrade. The file system structure is also used to store dialog input and system configuration values retrieved by the tool in order to allow the SUM to resume processing if it got interrupted for some reason.

The code for Java upgrades is mainly written in the Java programming language. Many XML control files are also present to define workflow parameters or to process dialog screens. The upgrade programs are supported by a small set of C/C++ programs and shared libraries or shell scripts to execute OS specific tasks. In contrast to SUM version 1.0, the archive for SUM version 2.0 does not contain the Java part, as there is no support for Java upgrades in that SUM version.

In addition to these parts, two other components are also always shipped as part of the upgrade tool: The SUM archive provides its own version of the SAP Java Virtual Machine (SAP JVM) to execute Java programs. Usually there is no need to apply patches to this JVM. SUM also does not integrate its own SAP JVM version to the SAP system to be upgraded. The second additional part is the SAP cryptographic library, which is needed to enable secured connections between the frontend GUI and the executing SUM process at OS level. In SUM version 2.0, the SAP cryptographic library does not have a directory of its own in that version but is integrated into the ABAP part.

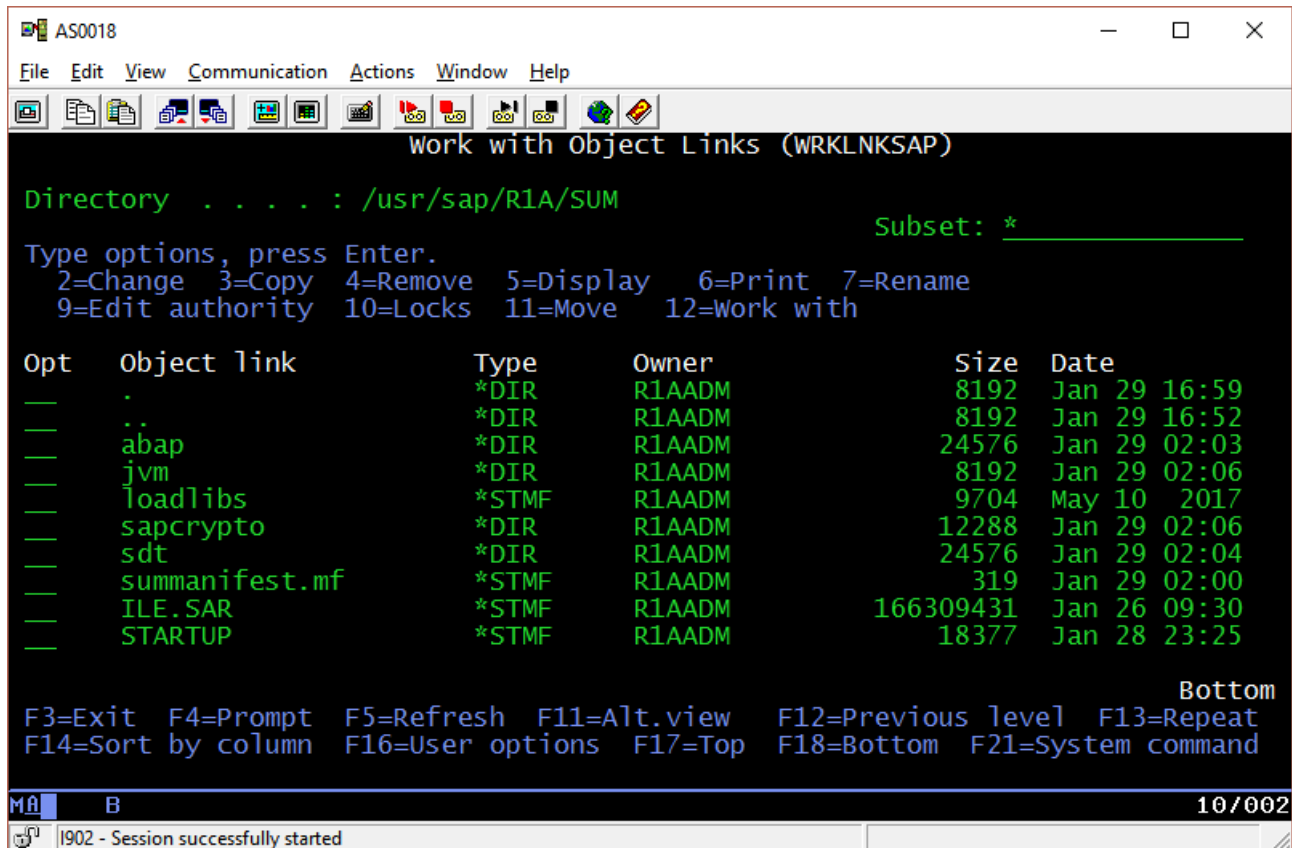


Figure 3: Directory structure of SUM version 1.0

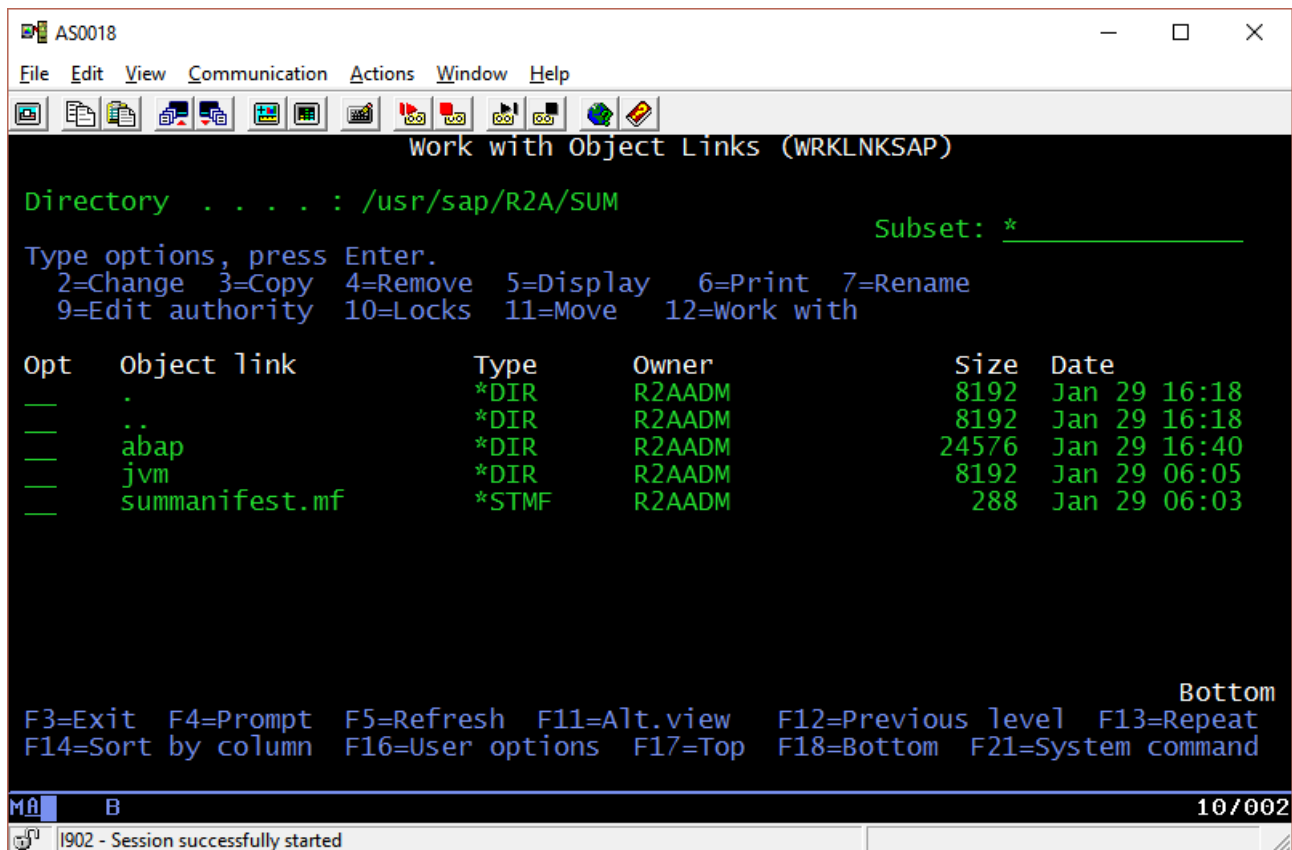


Figure 4: Directory structure of SUM version 2.0

## 3.4 Important File System Parts

### 3.4.1 Top-level Subdirectory Structure

SAP recommends extracting the tool archive `SUM.SAR` into the SAP system directory `/usr/sap/<SID>`, but any other directory in the Integrated File System (IFS) that is fully accessible by user `<SID>ADM` is allowed. To avoid subsequent authority issues, it is important to extract the archive only as user `<SID>ADM`. After extraction of the SAR archive, the file system structures shown in figures 3 and 4 can be found in the SUM root directory on the machine, depending on the SUM version.

### 3.4.2 Software for ABAP Upgrades

The directory `abap` contains several thousands of files, used for any kind of upgrade for SAP ABAP stacks. The SUM is able to configure itself, based on the SAP source release and the configuration of the planned SAP target release as described in the XML file that was delivered with the downloaded upgrade stack.

Besides the needed programs for the upgrade and their configuration files, the `abap` directory also contains ABAP transport files. For that reason, the directory structure looks like an ABAP transport directory and SUM uses it as such. During the runtime of the SUM, many additional files and directories are created. Therefore, the SUM directory can grow for several hundreds of GB, especially if SUM spans multiple release levels or includes a high amount of support packages. This needs to be considered for the upgrade disk space planning.

### 3.4.3 Software for Java Upgrades

The directory `sdt` is used for upgrades of SAP Java stacks. It consists also of the upgrade program parts (mainly Java archives with extension `”.jar”`), and a large set of XML files used for the upgrade control flow. This directory exists for SUM version 1.0 only. The dynamic growth of the directory size during runtime needs to be considered here as well.

### 3.4.4 SUM Specific Java Runtime

SAP provides a Java runtime (SAP JVM version 8) for both SUM versions in directory `jvm`. Usually, this part of the SUM directory does not need to be patched.

For the ABAP part of the upgrade tool, it is used in just a few cases. The most noticeable use case is the email notification application that can automatically send an email if SUM requires attention, in case a dialog needs to be answered by the user.

For Java upgrades (with SUM version 1.0 only), the whole upgrade process is driven by this JVM. Any external Java VM provided by the OS vendor, other third-party vendors, or the SAP system itself are not supported for the use with the SUM. If the SAP system to be upgraded contains a Java stack, the SUM also does not copy any part from the SUM specific Java version into the system or vice versa. If a newer version of the SAP system JVM needs to be installed during the upgrade, SUM only installs a version from the downloaded upgrade stack, which fits the planned SAP target release.



### 3.4.5 Cryptographic Software

To enable secured connections to the SUM processes and to read encrypted data from the corresponding SAP system, SUM contains the SAP Crypto Library. For SUM version 1.0, the subdirectory `sapcrypto` is used to enable the access to the ABAP and Java components of the SUM. For SUM version 2.0, the cryptographic libraries are integrated into the `abap` subdirectory. It does not make a difference for SUM whether the SAP system being upgraded is configured itself for use of the SAP Crypto Library or not.

### 3.4.6 Remaining Parts

The SUM root directory contains file `summanifest.mf`, which describes build and version information about the SUM archive itself. This file contains details to identify the exact SUM version, which is needed when reporting an upgrade problem to SAP in a customer incident. As already mentioned in chapter 3.1 *Overview*, there are different SUM versions available. Several times per year, SAP publishes new support packages of the Software Logistics Toolset that SUM is part of. Furthermore, SAP releases patches for each SP level to provide minor fixes and improvements. The order of information in the manifest file is slightly different between the SUM versions. The manifest file provides all important information in the following format:

SUM version 1.0	SUM version 2.0	Comment
Manifest-Version: 1.0 keyname: SUM keyvendor: sap.com keylocation: SAP AG  os: as400_pase_64 compilation mode: UNICODE compiled for: 64 BIT	Manifest-Version: 1.0 keyname: SUM keyvendor: sap.com keylocation: SAP AG  os: as400_pase_64 compilation mode: UNICODE  native branch: lmt_020 loadtools branch: lmt_020	Tool is always provided by SAP  ID is always <code>as400_pase_64</code> SUM is always built in UNICODE and for 64bit only  Source branch (in version 2.0)
release: 1.0 support package: 21 patch number: 7	release: 2.0 support package: 0 patch number: 10	SUM version, support package and patch level information
native branch: lmt_007 java branch: lmtj_007_REL		Source branch (in version 1.0)
assembly time: 2018-01-09 22:00:20	assembly time: 2018-01-07 23:02:20	Build time stamp of <code>SUM.SAR</code>
pack version: 54 pack tool version: 1.053	pack version: 13 pack tool version: 1.053	

In addition, there are two files specific to the IBM i platform in the SUM root directory for SUM version 1.0: the additional archive `ILE.SAR` and the shell script `loadlibs`. SUM version 2.0 also provides these two files, but there they are integrated into the `abap` subdirectory. Both files are related to the so-called upgrade tools library, which contains some ILE programs needed to run the upgrade. In contrast to former SUM versions, there is no longer the need to process these files manually before an upgrade. The files are now processed automatically during phase `PREP_INFRASTRUCTURE_OS4`. Besides other platform specific actions, this phase creates two libraries, library `SAP_TOOLS` and the upgrade tools library with a default name of `SAPSLT<SID>`. Both libraries get overwritten without any notification, if they already exist.



## 3.5 SUM running on IBM i Machines

### 3.5.1 Minimum OS Release

The minimum release of the operating system for the SUM depends on the underlying SAP kernel build release, which is different for the two SUM versions. This is a dependency from the build process when creating all the needed programs for the SUM archive. Other components, such as the SAP JVM runtime, may have their own restrictions. Besides this consideration, it is always recommended to stay current with the OS release level to avoid known issues and to benefit from the latest improvements. The following table shows the current requirements. SAP always recommends to keep the operating system of the IBM i server up-to-date. The lowest currently by IBM supported OS release is IBM i 7.2.

SUM Version and Use Case	Based on SAP Kernel Release	Minimum OS Release
1.0 for Java only and Non-UNICODE systems	7.21	IBM i V5R4 <sup>2</sup>
1.0 for ABAP and Dual stack UNICODE systems	7.21 and 7.49	IBM i 7.1 <sup>3</sup>
2.0	7.53	IBM i 7.2

### 3.5.2 Identifying SUM Processes

All SUM processes and their subsequently triggered jobs are running in subsystem QUSRWRK. For ABAP stacks, they are always called SUMSTART and run usually under control of the user profile <SID>ADM of the corresponding SAP system (see [figure 5](#) on page 17). Typically, they execute the program SAPup\_real, but also other programs can be active for different tasks. For Java stacks, the master job is called SLPROTOCOL and executing a Java program. In case of a dual-stack system, both types of jobs are active in parallel during the upgrade procedure.

Technically, all SUM processes are started by the SAP Host Agent. The SAP Host Agent worker job always runs with user profile SAPADM. After creating the new processes, they are switched internally to the required user profile <SID>ADM before the real upgrade program is executed. Therefore, this user profile name will be inherited by the SUM processes in their job names. That leads to qualified job names like <nnnnnn>/**SAPADM**/SUMSTART. As long the job is active, the effective user profile for that job can be found using option 1 (Display job status attributes) of command WRKJOB while the screen output of command WRKACTJOB always shows the effective user profile name a job is executed with.

### 3.5.3 Independent Auxiliary Storage Pool

Many customers run an IBM i partition in conjunction with an Independent Auxiliary Storage Pool (IASP) as part of their high availability solution. An independent ASP allows implementing a switchable storage area for the file system as part of an SAP system and its database. The upgrade tool

- 
- (2) SAP kernel release 721\_REL itself can be used with IBM i V5R3 or higher. Since SUM is always shipped together with an SAP JVM runtime version for internal purposes, it cannot run on V5R3. All released versions of SAP JVM require at least IBM i V5R4 (see [SAP Note 1367498](#) – SAP JVM installation prerequisites).
  - (3) For ABAP systems with a UNICODE codepage, SUM uses meanwhile a built-in tool based on the SAP Kernel release 7.49 to run SQL statements to modify database structures. This tool raises the minimum required OS release to IBM i 7.1.

supports such configurations without additional settings, if the SAP system has an IASP configuration as described in following SAP Notes:

- [SAP Note 568820](#) IBM i: Implementing an Independent ASP (iASP) System
- [SAP Note 1309481](#) iASP: Directory structure and system settings

In the most upgrade scenarios, a so-called shadow instance will be created. This is an almost full-fledged SAP instance, which is used to prepare database parts for the target release. As a fixed requirement, the database of the shadow instance must be created in the same ASP as the original database.

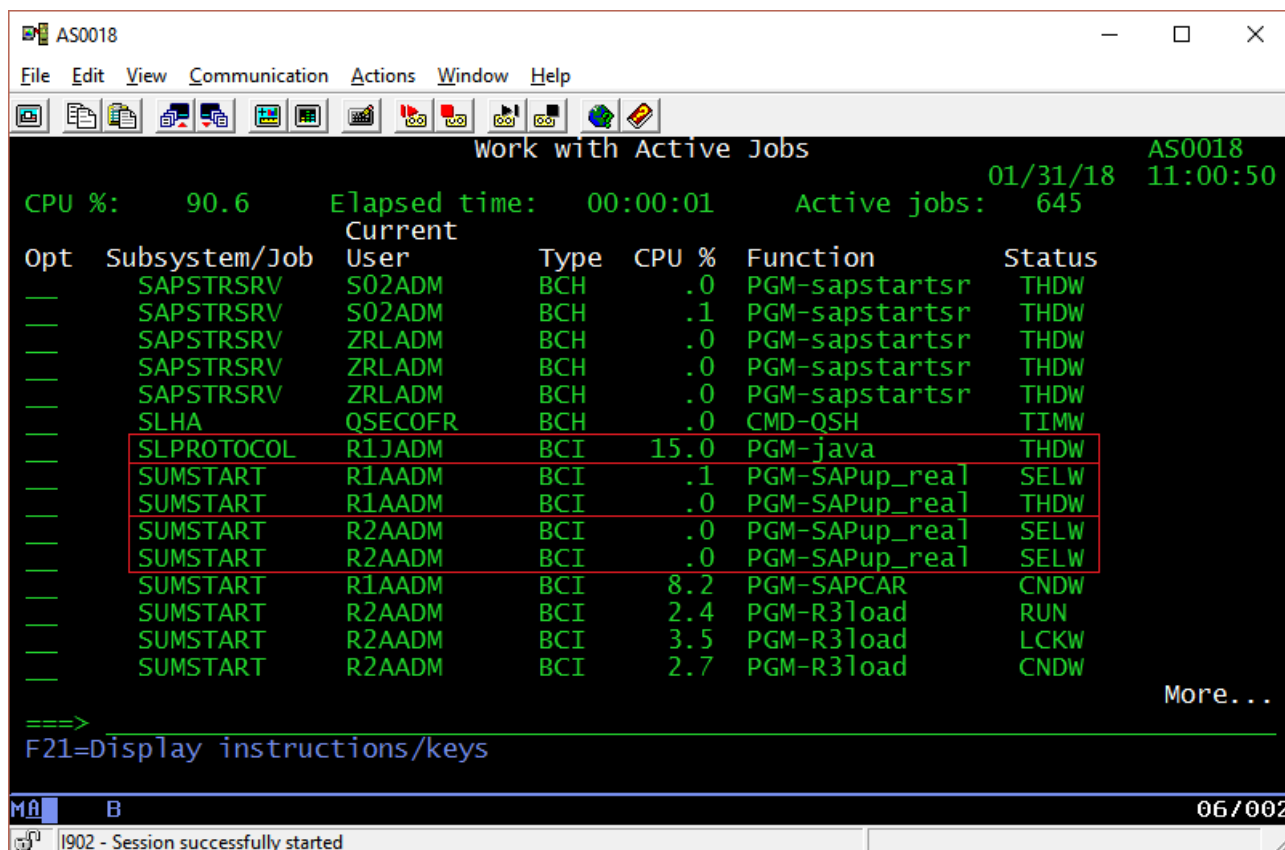


Figure 5: Running SUM processes in WRKACTJOB, overview with parallel SUMs for three SAP systems

## 4 Software Update Manager Processing in Detail

### 4.1 Starting and Stopping SUM

#### 4.1.1 Web Browser Access

Since the shipment of SUM 1.0 SP15 in 2015, the start of the SUM process is triggered by web browser access. There is no difference in the tool handling on IBM i compared to other SAP supported platforms. There is no need to configure a local web server on the IBM i server as this functionality is provided by the SAP Host Agent. At the writing of this paper, a minimum release level of SAP Host Agent release 7.21 patch level 25 must be installed. The host agent job also supports secured web access using the `https` protocol to provide encrypted communication between the SUM processes and the workstation running the GUI in a browser. The SUM guide contains a detailed description about the required configuration to enable the use of `https`. The upgrade process itself is started by entering a web URL in the browser of the workstation as documented in the upgrade guides. The TCP ports 1128 (with standard `http` access) or 1129 (with encrypted `https` access) are served by the SAP Host Agent worker job.

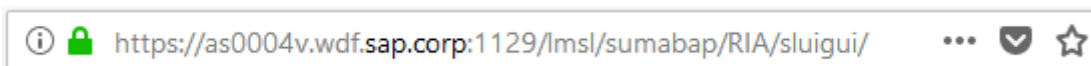


Figure 6: Browser address bar with active secured HTTP access

In some cases, the corresponding SUM process cannot be launched. The first check point in such cases is the standard logfile of the SAP Host Agent (`/usr/sap/hostctrl/work/sap-startsrv.log`). This file can show helpful information, if the SAP Host Agent is not configured correctly or not working properly. In addition, the following SAP Note provides a collection of already known issues and solution proposals if the SUM cannot be started on an IBM i server. This SAP Note is updated when new symptoms are found:

- [SAP Note 2300946](#) IBM i: Using Software Update Manager (SUM) to analyze start problems

#### 4.1.2 Running Multiple SUMs in Parallel

One of the biggest benefits of the IBM i platform is its capability to run several SAP systems in parallel in the same logical partition (LPAR). The concept of using subsystems allows the dedicated configuration of required hardware resources per SAP instance, such as the size of memory pools. With that, the CPU capacity is shared between the SAP systems by using automatic load balancing.

To fully support such landscapes, the SUM tool has been enabled to run independent upgrade procedures on separate SAP systems at the same time on the same machine. Former versions of the SAP upgrade tools required a special configuration, but two or more SUMs can be installed into different directories and executed at the same time without additional setup of the tool. When following SAPs recommendation to install the tool into directory `/usr/sap/<SID>/SUM`, each SAP system automatically get its own copy of the tool directory. A required preparation is the configuration of the SAP Host Agent to be able to work with the SUM on each of the SAP systems. It is important to choose a path to the SUM with the SAP system ID as the only variable part, like following examples:

```
/usr/sap/<SID>/SUM
```

```
/work/<SID>/SUM
```

The registration of a SUM directory at the SAP Host Agent is done with the following commands for each SAP system ID (after opening a PASE shell with the OS command `CALL QP2TERM`) with a privileged user profile like `QSECOFR`:

For SUM version 1.0: `/usr/sap/<SID>/SUM/STARTUP confighostagent <SID>`

For SUM version 2.0: `/usr/sap/<SID>/SUM/abap/SUMSTART confighostagent`

These calls generate configuration files in the SAP Host Agent directory to enable the communication between the tool SUM and the web browser on the used workstation. Once the files have been written, there is no technical need to run the scripts again – except if the SAP Host Agent itself needs to be updated. The generated files are not specific to a given SAP system ID but are used as templates by the host agent. The SAP system ID is part of the web browser URL to distinguish between the SAP systems when running more than one SUM at the same time on the same partition.

## 4.2 Using the SUM GUI

After switching the SUM GUI to the browser-based approach, many new features and utilities have been added to the GUI. SAP encourages the users of SUM to explore and to use them, and to report feedback about their experience to improve the GUI.

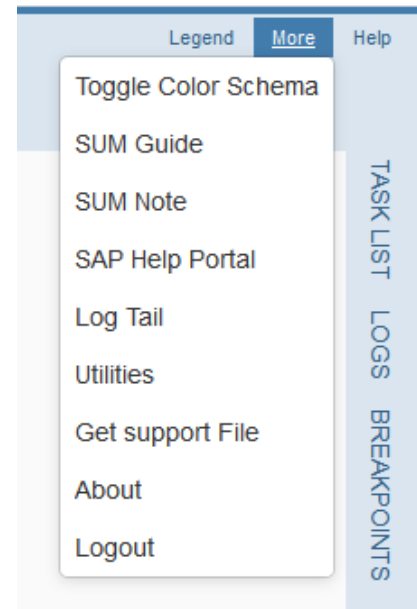


Figure 7: The "More" menu

### 4.2.1 GUI Options

In general, the GUI options are platform independent. The most useful features are briefly described here. Some of the features are shown very prominently, other functions are called by their respective menu items. In general, the SUM GUI provides a single point of access to all vital information related to the upgrade process. It can be easily managed within the browser session. Currently, most SUM GUI options are available for ABAP stack upgrades only.

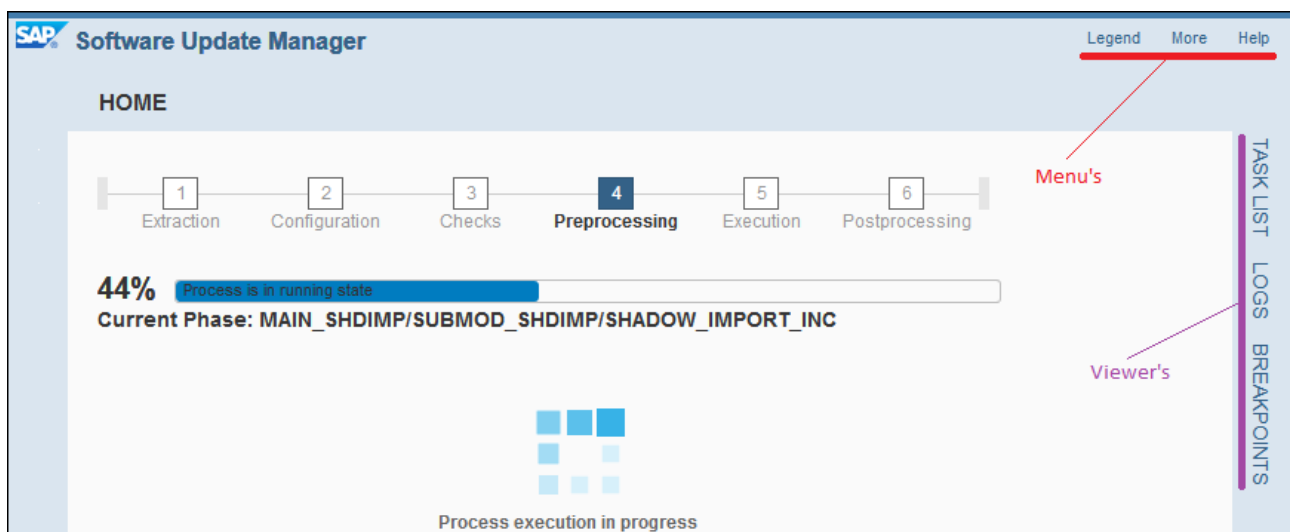


Figure 8: Progress view for ABAP upgrades and other GUI elements for SUM 2.0

When starting the SUM process, the GUI starts in the browser and switches automatically to the standard viewer to monitor the upgrade process and to show a progress bar. An important detail to know is what the percentage number means. It is not related to the past runtime, nor can it be used to calculate the remaining time until the SUM is finished. Instead, it represents the progress based on the number of phases to be executed in total. There is a difference in the behavior between ABAP and Java stacks. For ABAP stacks, the number of already processed phases is counted compared to the total number of phases for the complete SUM process (in percent), while for Java stacks, the phases get counted separately for each roadmap step (shown above from the progress bar). Estimating an exact time frame for SUM processes is very difficult to achieve. Usually the SUM runs in parallel to other workloads on the machine and uses parts of the hardware resources. In comparison to test upgrades of a copied SAP system, the runtime for the productive system may be different due to the possible variability of the workload or different hardware configurations.

If the SUM requires some input from the user, it switches automatically the screen and gives a brief description about the requested information. The color of the progress bar changes from blue to yellow to indicate that the user's attention is needed. Other used colors are red for error messages and green for finished milestones. When the SUM process is running, the currently executed upgrade phase is displayed, which is also an important information if contacting the SAP Support. The GUI also provides easy access to certain files and status information about the progress that has been made so far. On the right-hand side of the GUI is a vertical bar with links to the topics *Task List*, *Logs* and *Breakpoints*.

Description	Last Modified On
PHASES.LOG	2018.02.02 - 14:24:23
SAPUpStat.log	2018.02.02 - 14:24:23
SAPUp.ECO	2018.02.02 - 14:24:23
SAPK-75003INSAPABA.R1A	2018.02.02 - 14:24:23
ALOG750	2018.02.02 - 14:24:17
SAPK-74004INSTPI.R1A	2018.02.02 - 14:24:04
SAPK-75003INSAPGWFD.R1A	2018.02.02 - 14:23:39
SAPK-75002INSAPBW.R1A	2018.02.02 - 14:22:40
SAPK-75003INSAPBASIS.R1A	2018.02.02 - 14:20:09
SAPK-75003INSAPUI.R1A	2018.02.02 - 14:14:46
SAPK-74003INSTPI.R1A	2018.02.02 - 14:01:29
SAPK-75002INSAPABA.R1A	2018.02.02 - 14:01:11
SAPK-75002INSAPGWFD.R1A	2018.02.02 - 14:00:37
SAPK-75001INSAPBW.R1A	2018.02.02 - 13:59:05
SAPK-75002INSAPBASIS.R1A	2018.02.02 - 13:57:14
SAPK-75002INSAPUI.R1A	2018.02.02 - 13:51:08

Figure 9: SUM for ABAP stacks with active log file viewer

The task list shows the complete phase list that has been calculated for the given upgrade scenario, based on the definitions provided by the SAP development. This is not necessarily the same list of phases that are actually executed. Some of the phases in the definition scope are intended to be executed for certain OS or database types only, or, for example, for dual-stack systems but not for single stack systems. The task list viewer shows the list of all defined tasks together with an icon, indicating what happened with them. The menu *Legend* provides a short explanation of each possible icon.

Another useful feature of the GUI is the embedded viewer to read the files directly from the `SUM_log` subdirectory on the IBM i server. The list of log files is sorted by time in descending order and shows the newest log files first. Besides the plain log file list, each file is represented as a link in order to read its content. [Figure 9](#) on page 20 shows the GUI with an active log file viewer, which appears on the right side of the same browser window.

The *Breakpoints* viewer works similar to the other integrated viewers. It lists all the upgrade phases together with a checkbox icon. If the checkbox is selected, the breakpoint is defined and active. Multiple breakpoints can be defined in parallel, and each one causes a separate stop. The breakpoint dialog provides different options regarding how to proceed: deleting all the breakpoints, continuing to the next breakpoint or switching into *Single Execution Mode*, where SUM stops as it enters each next phase. Active breakpoints in already processed phases no longer influence the SUM procedure. It does not matter whether they are kept or removed. Working with breakpoints requires to have enough knowledge about the SUM procedure and the sequence of long running phases. Finally, the time line of the SUM project needs to be planned and adopted accordingly for the use of breakpoints.

The GUI provides additional tools and links to provide information about and to analyze the running SUM process. They are collected in the menu *More* in the top right corner of the GUI. The links to the upgrade documentation are most important. These links are aligned with the currently used SUM version and support package level and point to the right content on the SAP support web site. During the SUM procedure, different evaluation phases are executed. They collect runtime information and detected system settings. SUM generates reports based on the collected information for viewing in the browser as well. Those reports can be accessed using the menu *More*, topic *Utilities* (as shown in [figure 7](#) on page 19). This link opens a new browser window with its own navigation to provide access to the tools included in the SUM. Currently, these utilities are only available for ABAP stack upgrades.

During its runtime, SUM stores any input values from dialogs as parameters in a repository file. This file cannot be edited directly due to security and data integrity concerns, but it may be necessary to modify its contents during a running SUM process. One scenario would be to correct a changed *DDIC* password (e.g. locking user *DDIC* accidentally by entering a wrong password for three times) as SUM uses this password frequently for RFC calls. The password is stored in the repository, but always encrypted. You can resolve such a scenario by using the *SUM Parameters* review tool to check and modify all entered dialog values as well as the automatically calculated parameters.

The *SUM Phaselist* tool gives more detailed technical insight into single phases. It starts with an overview of the SUM run modes, and you can drill from the roadmap sections down to single phases. The list contains a short description, the used logfile name and information about necessary user actions. Some phases show the entry `auto` as logfile name. This means, file `PHASES.LOG` is used as logfile, and those phases have usually short runtimes. To avoid losing any logging information, updates to this logfile are copied to the end of file `PHASES.SAV` before the next of these phases is started.

An interesting tool in this area is the *SUM Analysis* tool. As shown in [figure 10](#) on page 22, it provides runtime information of already executed upgrade phases. The page navigation allows one to drill from top level upgrade modules down to single phases to exact identify the most time-consuming parts of the SUM procedure. The *highlight* buttons can be used to identify phases which ran longer than a



given number of minutes. If long run times are caused by wait times due to unanswered dialogs in the GUI, they are marked accordingly. Finally, this tool also shows the number of repetitions and failures. Some phases are executed automatically twice by SUM by design.

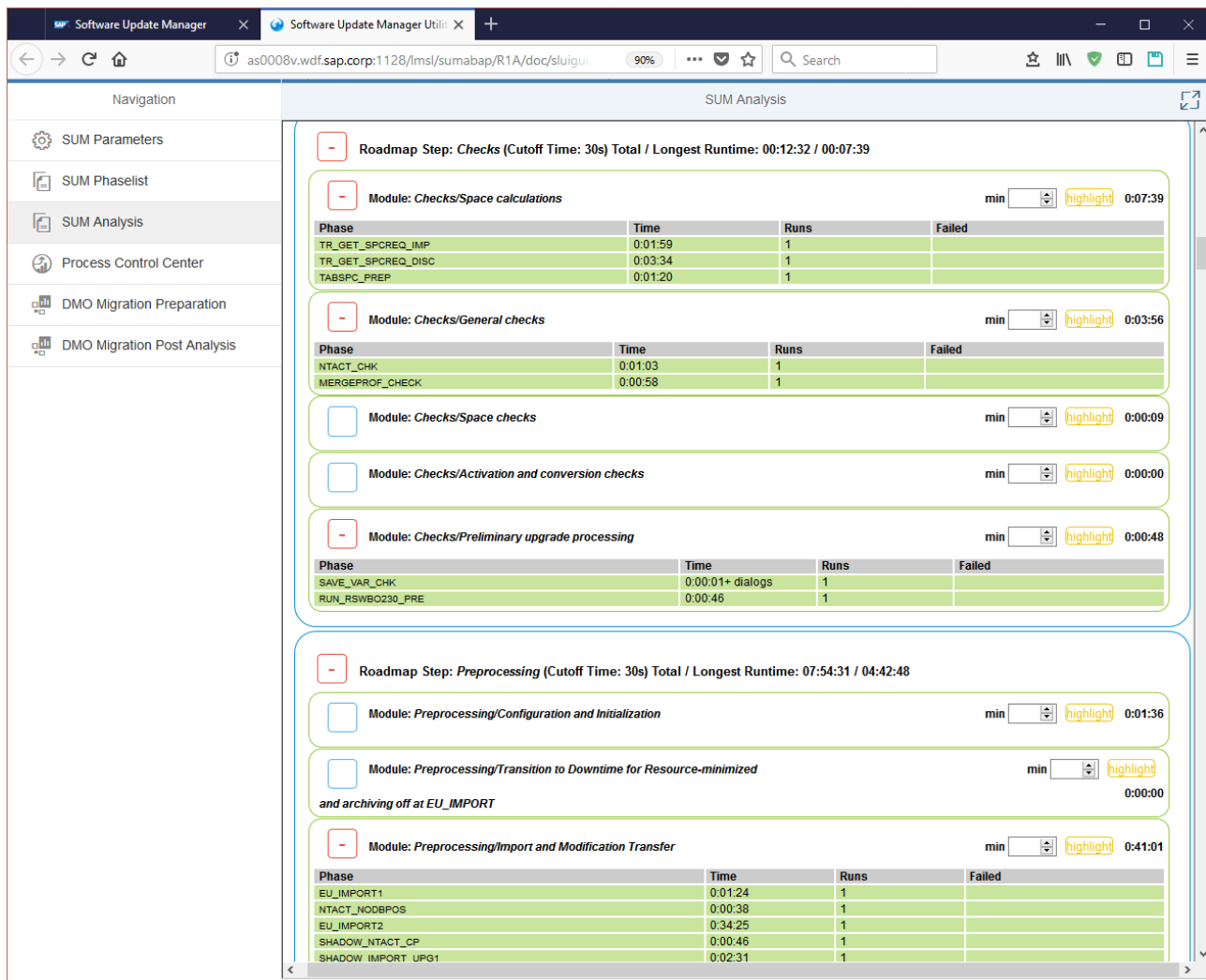


Figure 10: SUM Analysis viewer for ABAP upgrades

SUM may run programs in parallel (e.g. R3load or R3trans) in various phases to optimize the phase runtime, to distribute workload, and to use hardware resources more efficiently. The *Process Control Center* is a monitor that shows the degree of parallelism of the currently running upgrade phases. The diagram in [figure 11](#) on page 23 shows the number of active processes measured over the phase runtime. Usually, the dialog in phase INITSUBST asks the user to specify a maximum number of parallel processes. These values are used as defaults for the SUM procedure. The vertical slider on the left can be used to adjust the number of processes on the fly, when possible. When modifying the number of processes directly during the SUM runtime, sufficient hardware resources must be available. Otherwise, this can have an impact to the overall system performance of the server. After the modification, the SUM starts newly enabled parallel processes as quickly as possible, if the prepared upgrade data packages can be distributed to them usefully.

Finally, there are tools available to prepare, monitor and analyze workload related to the Database Migration Option (DMO). This option allows one to combine an upgrade with SUM and a migration



of the database to a different database management system (e.g. SAP HANA or Oracle). This kind of upgrade is more complex due to the switch of the database vendor, so it is currently not covered in this document.

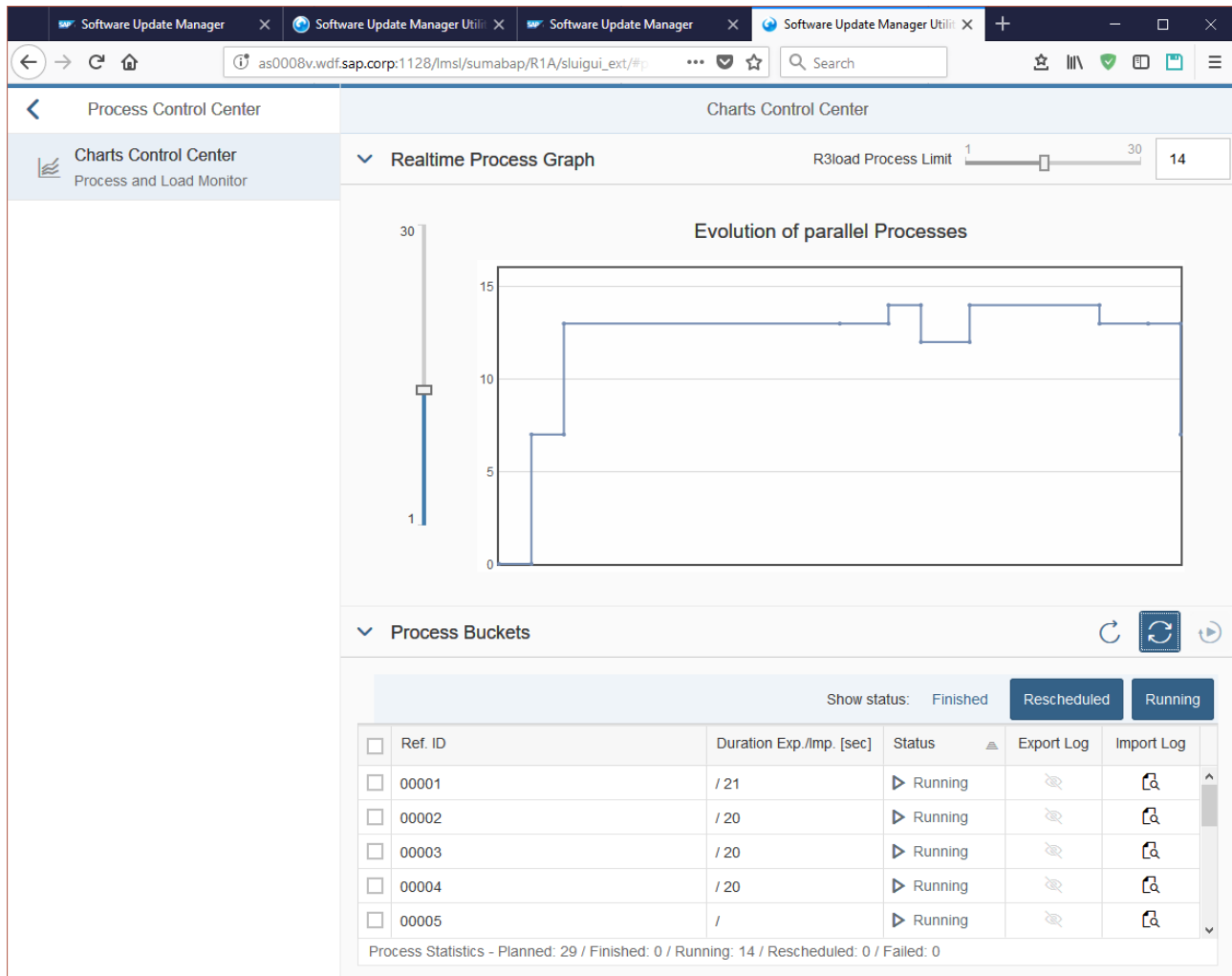


Figure 11: Process Control Center in a separate browser window

### 4.2.2 Roadmap Sections

The process of running SUM is logically split into several parts. The SUM documentation describes them as *Roadmap Steps*. The sequence is always shown on top of the GUI, and the current step is highlighted (see [figure 12](#)).

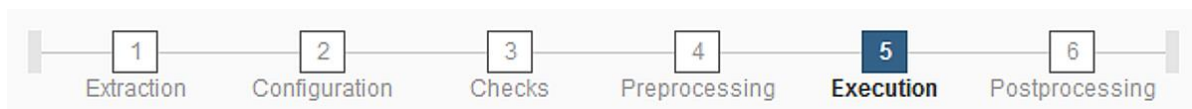


Figure 12: Roadmap steps in general

After every roadmap step, the SUM for ABAP stacks stops and shows a summary about necessary actions to be executed by the user, independent from the SUM procedure. The SUM for Java stacks

automatically switches between the roadmap steps, except for the switches before and after module *Execution*. If a Java stack is involved, an additional pre-execution roadmap is launched at the beginning (see [figure 13](#) on page 24). At the end of the pre-execution roadmap, the GUI switches automatically to the standard roadmap.



Figure 13: Pre-execution roadmap if a Java stack is involved

The roadmap steps are managed separately and in a slightly different way for ABAP and Java stacks, but their names and behavior are identical for both stack types. In dual-stack systems, each stack has a roadmap of its own, and both SUM parts run in parallel with synchronization phases to ensure that the SUM parts do not get decoupled. In that case, the GUI provides a tab-based layout to easily switch between the two roadmaps.

The first roadmap step in any case is the *Extraction* module. Most of the effort here is spent on the verification of all data files to be processed during the upgrade. This consists of the files in the so-called upgrade download stack and of physical media such as DVDs or Blu-ray disks, if available. In addition, some standard checks are executed at the beginning to verify that the SAP system can work with the given SUM version, and that the SAP system is ready for the use of SUM. For ABAP stacks, any upgrade phase with the word SCAN in the name (like SCANDIR\_EPS0) is time-consuming. Phases for the tool import (phase names starting with TOOLIMP...) also take some time. In general, this first module is usually done in less than 2 hours. The runtime depends on the number of files or physical media included in the upgrade. For Java stacks, this roadmap step is done very quickly as most of the file checks are done either in the pre-execution roadmap or later in the next module.

The *Configuration* step includes a number of user dialogs. For ABAP stacks, it starts with upgrade phase INITSUBST to ask for the number of parallel processes to be executed later in different phases. Other parameters for the SUM are defined in that phase as well. Chapter 4.4 *Important Phases for Technical Decisions* explains this phase in more detail. There are just a few important phases regarding the runtime, for example phase EHP\_INCLUSION, or phases running ABAP check reports, like the phase CONFLICT\_CHECK. The runtime of the roadmap step *Configuration* can vary considerably. Most of the checks run very fast. The workload directly depends on the size of the provided download stack and on the number of files it contains for both ABAP and Java stacks. Most of the delays during this step are caused by dialogs where the SUM simply waits for input. These wait times can be optimized by using the *SUM MailNotifier* as described in the SUM guides, so that notifications get sent out directly to inform about waiting dialogs by an email.

In general, additional time buffers must be planned for actions required by the SUM itself when running phases in roadmap step *Checks*. Well maintained SAP systems do not have problems to pass the check phases, but if checks are failing, the SUM usually runs the complete roadmap step again to verify results from already passed checks.

Step *Preprocessing* is the most time-consuming part of the SUM. In this step, the shadow instance is created if needed, and its corresponding shadow database. A few scenarios are working without a shadow instance (e.g. when applying just a support package stack), but a release upgrade or an enhancement package stack implementation require such a temporary instance. More information about the shadow instances can be found in chapter 4.3 *Upgrade Shadow Instances*. For ABAP stacks, there are several long running upgrade phases (e.g. SCEXEC\_ALIAS, DDIC\_UPG, ACT\_UPG or

SHADOW\_IMPORT\_INC) that may run several hours, depending on the upgrade scenario and its complexity in terms of included add-ons or support packages. For Java stacks, the longest running phase is typically IMPORT-JLOAD. Once all preprocessing steps have finished, SUM stops and the user should perform a complete backup of the SAP system, including the SUM directories. This is the latest point in time to which the SUM procedure can be easily reset by restoring the backup, and modifications can be reverted without affecting the users in the SAP system.

The *Execution* step is also called upgrade downtime. In this section, the SAP system will be shut down to exchange the kernel executables or other SAP system runtime components, such as the SAP JVM. After the new kernel binaries have been installed, the SAP system is started and stopped several times for purposes that require the complete switch of the SAP infrastructure to the given target release. SAP users cannot login during these short uptime phases, except administrative users like *DDIC* in ABAP or *administrator* in Java. There are some typically long running phases like SHADOW\_IMPORT\_INC, PARCONV\_UPG, TABIM\_UPG or XPRAS\_AIMMRG for ABAP stacks and the different deployment phases for Java stacks. At the end of the *Execution* step, the SAP system will be up and running. This is the first moment when the SAP system has been switched completely to the new release. The SUM suggests another backup at this point. For this backup, it makes sense to shut down the SAP system once more to avoid lock situations. But since all of the upgrade steps have not yet completely finished this final backup is often postponed to the end of the *Postprocessing* module. At least for productive SAP systems the backup strategy has to be followed as described in the upgrade guide.

**Feedback to SAP**

What kind of system did you run the process on? 
 Development system  
 Quality system  
 Production system  
 Actual system  
 Test on copy of the system

Did you run the process on the actual system or a copy of it (i. e. test process)?

How many times did you contact SAP SE for help with your process (customer calls or incidents)?

Did the process downtime meet your expectations?  Yes  No

Did the process documentation meet your needs?  Yes  Partly  No

Can we contact you in case of questions? If yes, please enter an e-mail address:

Contact Email Address:

Do you have any comments on the tool or process?

**Process XML:**

```

<![CDATA[<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="UpgAnalysis.xsl"?>

<!-- File generated with SAPup version 53.000 date 2018/02/28 11:24:09 -->

<analysis SID="R1A" instnumber="██████████" startdate="20180226144702"

```

Figure 14: SUM upgrade evaluation form with phase runtime XML

The SUM runs the last actions in roadmap step *Postprocessing* to finish the procedure. In general, the *Postprocessing* module runs in much shorter time than the previous modules. In case of a Java stack,

this step takes just a few minutes, while for ABAP stacks, many ABAP reports will run. Their individual runtimes are not that long, but the number of reports in total requires a certain amount of time. At the end of the upgrade, all effective phase runtimes are known and SUM calculates the final evaluation form ([figure 14](#)). If you want to support SAP in optimizing the SUM upgrade runtime, it would be helpful to send in this form. There are optional fields to enter some SAP system details, an email address and an additional comment, but the important part is the evaluation XML file generated by the SUM, which contains the phase runtime information for the complete upgrade procedure.

### 4.2.3 User Defined Breakpoints

Breakpoints can be used to stop the SUM process at a given phase for various reasons. The behavior is like normal dialogs during the upgrade procedure, but the user can choose the breakpoint himself. One example of when one may want to set a breakpoint is for the purpose of manual resource balancing during the *Preprocessing* roadmap step. This roadmap step contains many long running upgrade phases (e.g. phase `SHADOW_IMPORT_INC`), but only very few dialogs where the procedure stops to allow user input. It also runs in parallel to the SAP system, while users are active and run their daily work load. Typically, 50% or more of the total SUM runtime is spent in these *Preprocessing* steps. One might want to keep the SUM from starting such a long running and resource consuming phase while the daily business still requires a certain amount of resources. Having a breakpoint defined causes SUM to stop before the phase gets started. While SUM is waiting in the breakpoint dialog, machine resources can be used for other tasks.

## 4.3 Upgrade Shadow Instances

SAP upgrades drive many and complex changes to the SAP database. In addition, new database content related to the target release is loaded into the tables. Because an SAP database contains several tens of thousands of tables, this takes a certain amount of time. While this is going on, the SAP programs and data structures must remain in sync at the same release level to for SAP users to continue working as long as possible. During the upgrade procedure, there is a point of time where all the users have to log off from the SAP system and the upgrade can switch to the target set of programs and data structures. The need to introduce new programs and structures conflicts with the customers' need to have the SAP system active – in the best case 24 hours a day and 7 days a week. To solve this conflict and to reduce the needed downtime, SAP already introduced in early versions of upgrade tools (the predecessors of the SUM) a mechanism called *Shadow System*. A good high-level description about the shadow system is listed in the SAP Blog entry [SUM: introduction to shadow system](#). It describes the concept of creating an additional SAP system in parallel to the existing one, to be used for database modifications, while the SAP users stay active on their system as long as possible. The required time span where the SAP system is unavailable to users is not gone, but it is now significantly shorter.

### 4.3.1 Shadow Database Libraries

The ABAP shadow database can be described as a logical database, as it does not contain physical tables (objects of type `*FILE-PF`), but it provides full SQL access to all tables. For that, the SAP kernel programs and tools use C/C++ programs with an interface provided by SAP to send SQL statements to the database. A sample view to a shadow database library is shown in [figure 15](#) on page 27.

When taking a closer look at such a shadow table with the `DSPFD` command as shown in [figure 16](#) on page 27, it becomes obvious how SAP is working with those tables. The command lists a lot of

information about table ACC\_I00001. Technically it is an SQL view, which is a logical file on IBM i (object of type \*FILE-LF). Since object names on IBM i have a maximum length of 10 characters, an automatic mapping mechanism takes place to allow the use of longer SQL names as described in the SQL standards. In the output of the command DSPFD, you can see the alternative name, the real SQL name, which in this example is ACC\_INPH3\_DPR.

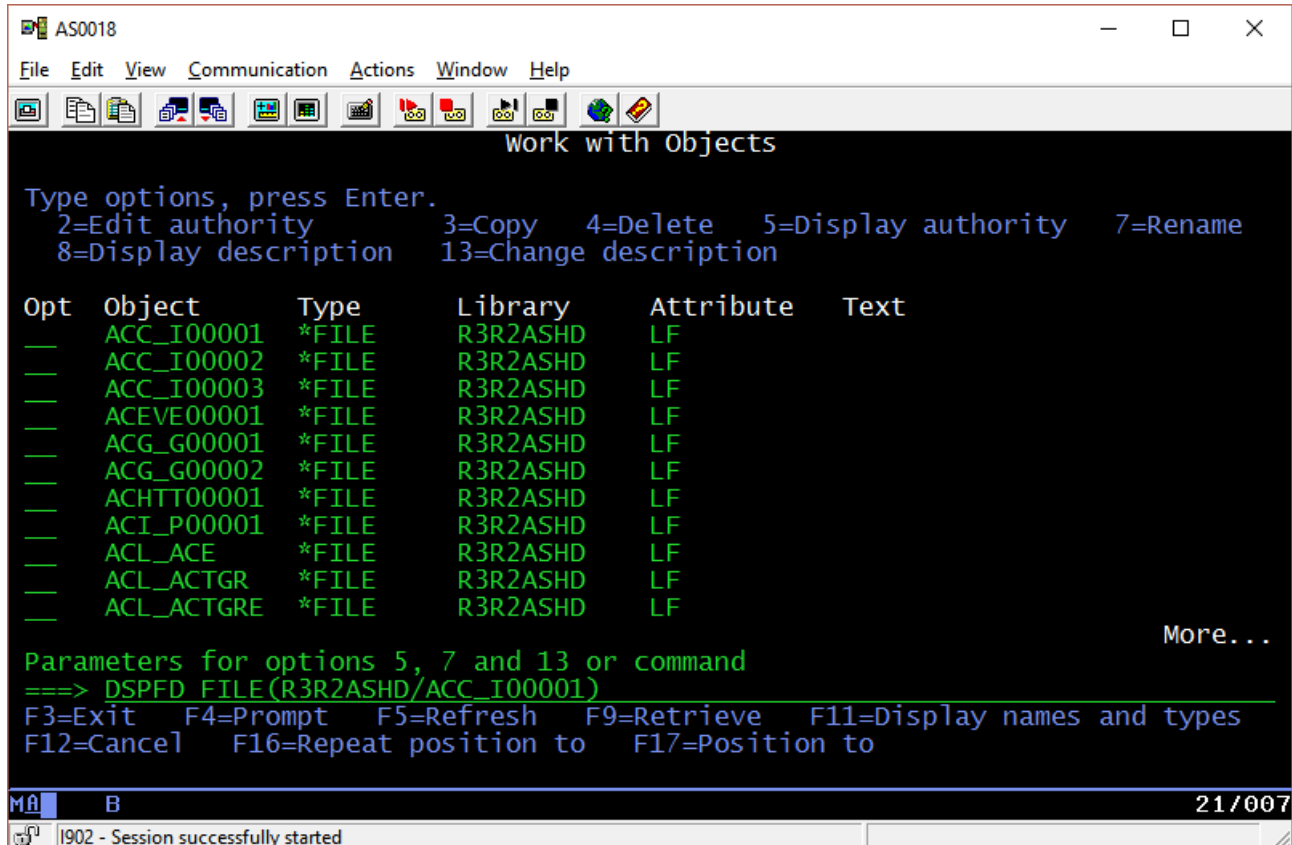


Figure 15: ABAP shadow database library with views to shadow tables

The command also shows the SQL statement that created this view. If command DSPFD is executed for the base table R3R2ADATA/ACC\_50004 (which is an object of type \*FILE-PF) in this statement, the SQL name of the real shadow table is shown as "ACC\_INPH3\_DPR?" (figure 17).

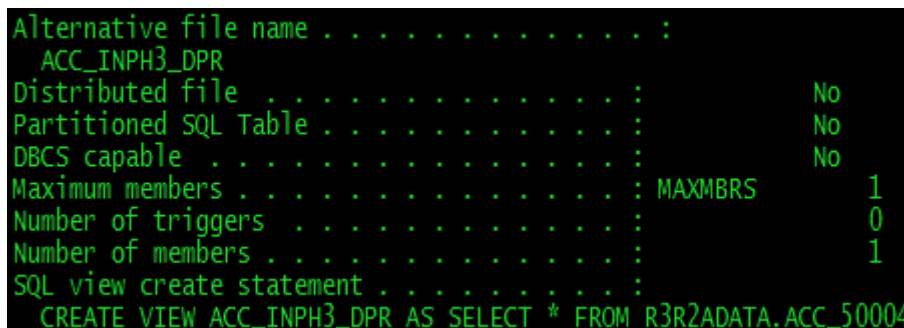


Figure 16: Output of DSPFD for shadow view



Figure 17: Real name of the shadow table

The question mark character at the end of the name is specific to the implementation on IBM i and denotes the table as shadow table. Other DBMS use different special characters to distinguish between standard and shadow table names, but the concept is the same. When accessing shadow tables with SQL, quotes are necessary to avoid syntax errors because of the special characters.

Char	Example	Comment
?	ADRC?	A shadow table stored in the standard database library. They are usually not known to the data dictionary and therefore listed in the output of transaction DB02 (in the consistency check) when executed during upgrade roadmap step <i>Preprocessing</i> .
+	CWBNTDATA+CWB	A standard index name for a table. The naming convention for indexes in the ABAP database always uses a structure with 3 parts: the name of the underlying table, the special character as name delimiter and up to 3 characters as suffix.
&	ADRC&I01	An alternative index name for a table. The same naming convention is used as for standard indexes, but a different delimiter.

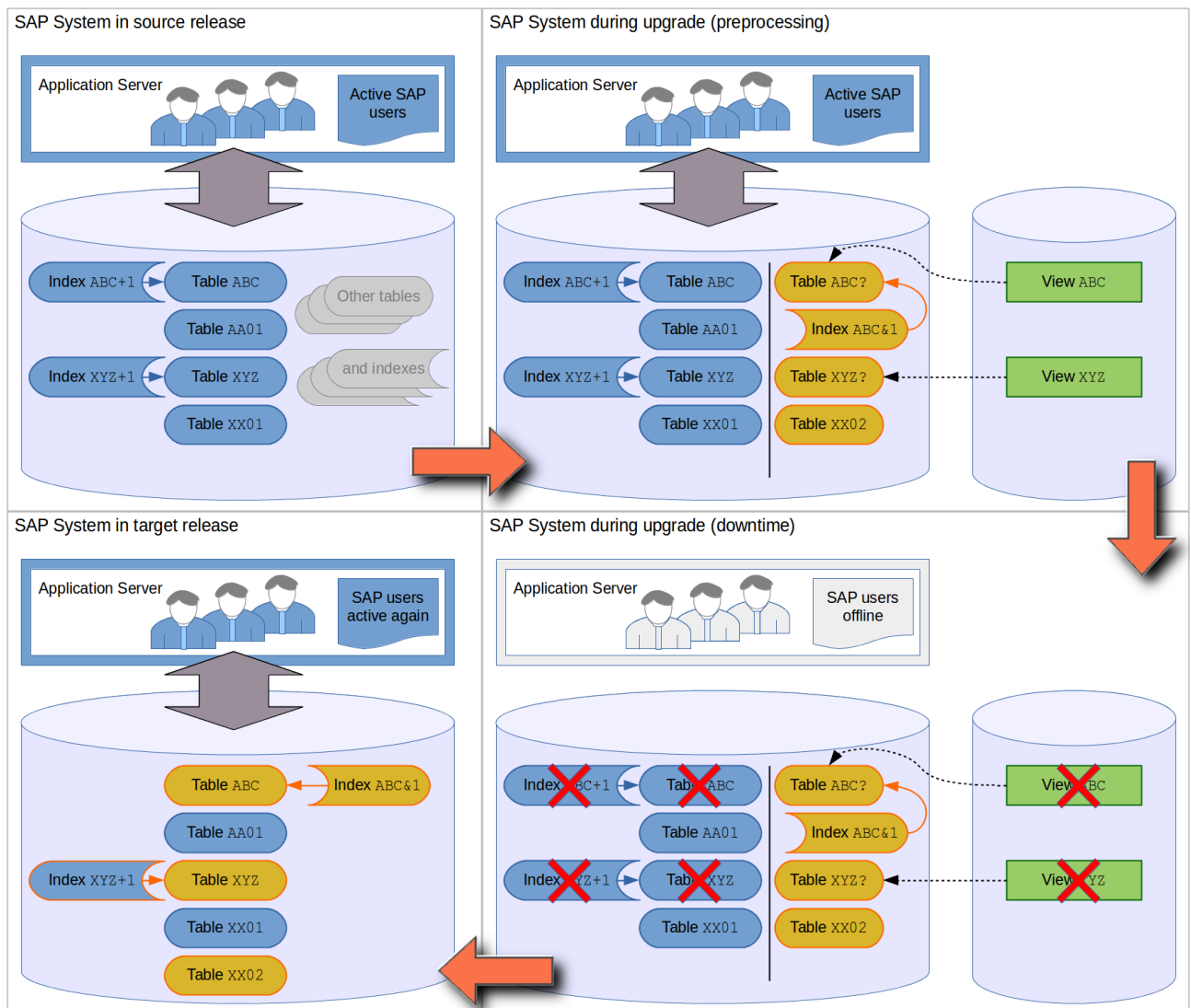


Figure 18: Modifications in the SAP database during the upgrade with a shadow database



There are a few special characters used in table names of SAP software, which have a special meaning in databases of an ABAP stack as shown in the table above. The special characters used in index names may switch from one to the other. In case the SAP system was upgraded several times, a mixture of name delimiters for indexes can be found in the database, but this is not considered an error. For example, the first upgrade creates a shadow table (a table name with question mark '?' at the end) for structural changes like adding a new data field. In addition, an index for that shadow table is created. The upgrade tool tries first to use the ampersand character '&' within the index name. If there is no name conflict, the request will succeed. Later during the upgrade process, the original table will get dropped together with its original index. Then the shadow table will be renamed back to the original table name and the question mark is removed from the table name, but the index name is not touched. A rename of this index would require the recreation of it, but this is a time-consuming task and therefore avoided. If the next SUM process is started, a new shadow database will be created, and for some indexes a name conflict may occur. In such cases, the upgrade procedure automatically switches back to the plus character '+' for the index name. This is the reason for the possible mixture of special characters in index names. [Figure 18](#) shows an overview, how tables and indexes are handled during the upgrade, especially how shadow tables replace their predecessors.

The screenshot shows the SAP AS0018 'Work with Objects' interface. It displays a table of database objects with columns for Opt, Object, Type, Library, Attribute, and Text. The objects listed include SYSVIEWS, T\_CHUNK, T\_DOMAIN, T\_SESSION, TAGOB00001, TAGOB00002, TC\_WD00001, TC\_WD00002, TC\_WD00003, TC\_WD00004, and TC\_WD00005. The text column shows 'SQL catalog view' for SYSVIEWS. Below the table, there are options for editing authority, copying, deleting, displaying authority, renaming, displaying description, and changing description. Parameters for options 5, 7, and 13 are also shown.

Opt	Object	Type	Library	Attribute	Text
—	SYSVIEWS	*FILE	SHDR1JDB	LF	SQL catalog view
—	T_CHUNK	*FILE	SHDR1JDB	PF	
—	T_DOMAIN	*FILE	SHDR1JDB	PF	
—	T_SESSION	*FILE	SHDR1JDB	PF	
—	TAGOB00001	*FILE	SHDR1JDB	LF	
—	TAGOB00002	*FILE	SHDR1JDB	LF	
—	TC_WD00001	*FILE	SHDR1JDB	DDMF	
—	TC_WD00002	*FILE	SHDR1JDB	DDMF	
—	TC_WD00003	*FILE	SHDR1JDB	DDMF	
—	TC_WD00004	*FILE	SHDR1JDB	DDMF	
—	TC_WD00005	*FILE	SHDR1JDB	DDMF	

Figure 19: Java shadow database library with different table types

The Java shadow database is different as the library contains a mix of different objects of type \*FILE. The Java shadow library is created in the same way as the ABAP shadow database library. After creation it is empty, except for the usual system catalog views. [Figure 19](#) shows a small part with SYSVIEWS as one of the catalog views on top. Some of the shadow tables are created as physical files (SQL tables) in the shadow database library. They are needed to run the shadow instance with data independent from the original instance. The logical files are typically indexes to those shadow



tables, or SQL views like the catalog views. Finally, the objects of type \*FILE-DDMF have the same function as the shadow views in the ABAP database. They point to tables in the original database schema with the prefix “SHD\_“. Due to the database access using the JDBC driver, a different technical mechanism is used for the implementation (see [figure20](#)).

```

File Description Header
File . . . . . : FILE      TC_WD00001
Library . . . . . :         SHDR1JDB
Type of file . . . . . :         Device
Device type . . . . . :         DDM
Auxiliary storage pool ID . . . . . :         00033
DDM File Attributes
File level identifier . . . . . :         1180315150608
Creation date . . . . . :         03/15/18
Remote file . . . . . : RMTFILE  'SAPR1JDB/SHD_RR_FGUID'

Remote location          RMTLOCNAME
Name or address . . . . . :         '*LOCAL'
Type . . . . . :         '*SNA'

```

Figure 20: Correlation between a shadow view (as DDM file) and the table in the original database

When running the DSPFD command against a Distributed Data Management (DDM) file, it shows the used mechanism for such tables. Because a DDM file can establish a connection to a different and even remote table, the DDM file TC\_WD00001 in the shadow database SHDR1JDB is pointing to the real physical table SHD\_RR\_FGUID in the original database SAPR1JDB. The technical setup is different compared to the ABAP shadow database, but the purpose is the same. It provides a certain name in the shadow database as an alias to a table in the original database, which is currently not known to the SAP dictionary. This mechanism avoids the time-consuming move of tables or their data after the shadow database gets deleted at the end of the preprocessing phases.

There are just a few upgrade scenarios that do not need to create a shadow database. For ABAP systems, a shadow is not required when SUM is used to apply a support package stack. The first dialog in upgrade phase INITSUBST provides an option to enable such a setup, if allowed. If the support package stack jumps to a significantly higher patch level or if many customer-modified ABAP reports are involved, it is recommended to use a shadow instance. In both cases, a large number of files must be processed. The overall upgrade process with a shadow instance takes a longer runtime in total but provides a higher flexibility and allows the SAP users to stay active longer in the system.

The lifetime of shadow instances and their related shadow databases are limited to the *Preprocessing* roadmap step of the upgrade process.

Whenever a shadow instance is used by the SUM, it requires to have some SAP specific resources available on the IBM i server. First, it requires to have an unused SAP instance number available and all TCP ports related to that number. The most important port numbers in relation to the SAP instance number are shown in the table below. Other ports might be in use as well while the instance is active. Some of the ports can be reconfigured to different port numbers using SAP profile parameters. The complete list with more details can be found in the SAP Help Portal article [TCP/IP Ports of all SAP Products](#). For application server instance types, a separate service instance is usually installed (ASCS<nn> for ABAP or SCS<nn> for Java, where <nn> is the 2-digit instance number). If present, they typically contain a standalone enqueue server and a message server. The SCS instance for Java also contains the gateway reader.

ABAP instances		Java instances	
TCP Port	Process / Program	TCP Port	Process / Program
12<nn>	RSLGSEND (if configured)	32<nn>	Java enqueue server enserver
14<nn>	RSLGCOLL (if configured)	33<nn>	Gateway reader gwrdr
15<nn>	RSLGCOLL (if configured)	39<nn>	Message server msg_server
32<nn>	ABAP dispatcher disp+work	80<nn>	IceMan icman
33<nn>	Gateway reader gwrdr	81<nn>	Message server msg_server
36<nn>	Message server msg_server	4<nn>00	IGS multiplexer job igsmux_mt
39<nn>	Message server (secured, if configured)	4<nn>01	IGS port watcher job igspw_mt
47<nn>	Dispatcher (secured, if configured)	4<nn>02	IGS port watcher job igspw_mt
48<nn>	Gateway reader (secured, if configured)	4<nn>80	IGS multiplexer job igsmux_mt
80<nn>	IceMan icman	5<nn>00	App. Server Java using icman (HTTP)
81<nn>	Message server msg_server	5<nn>01	App. Server Java (HTTP over SSL)
96<nn>	XDN listener job XDNCNTROL	5<nn>02	App. Server Java (IIOP initial context)
4<nn>00	IGS multiplexer job igsmux_mt	5<nn>03	App. Server Java (IIOP over SSL)
4<nn>01	IGS port watcher job igspw_mt	5<nn>04	Application Server Java (P4 port)
4<nn>02	IGS port watcher job igspw_mt	5<nn>07	Application Server Java (IIOP)
4<nn>80	IGS multiplexer job igsmux_mt	5<nn>08	Application Server Java (Telnet)
5<nn>13	Instance service job sapstartsrv	5<nn>13	Instance service job sapstartsrv
		5<nn>14	Instance service job sapstartsrv
		5<nn>20	Java main server job server0

Other relevant resources for shadow instances are placed in library R3<SID>400, objects such as job queues (object type \*JOBQ), job descriptions (\*JOBQ) and subsystem descriptions (\*SBSD). Those objects are kept for the whole upgrade runtime, in case they are needed for different purposes. If the shadow instance number is used later for any other purpose after roadmap step *Execution* has started, it is **no longer considered a shadow instance** - except for the near Zero Downtime Maintenance (nZDM). Currently, there are two use cases:

- Upgrades of dual-stack SAP systems:

During upgrade phases in roadmap step *Execution*, the dual-stack central instance gets temporarily split into two separate instances. The original instance is used as Java single stack instance, while a new single stack ABAP instance is created in parallel. Correspondingly, there are two directories DVEBMGS<nn> in directory /usr/sap/<SID>. For the temporary ABAP instance, the former shadow instance number is reused. This allows both parts of the SUM to operate independently from each other on the respective instances, especially to stop or start them during the downtime phases. At the end of the upgrade downtime, the temporary split is reverted automatically and both stacks get merged to operate again as dual-stack configuration, using the original instance name. The directory of the second DVEBMGS<nn> instance, using the former shadow instance number, gets removed later. With ABAP upgrade

phase `STARTSAP_PUPG`, the final start of the instances provides the dual-stack configuration again before entering the roadmap step *Postprocessing*.

- Creating a missing ASCS instance:

If the SUM detects an ABAP stack that has not previously been updated to use an ASCS instance, it asks in phase `ASCS_ASK` if the missing instance should be created. If this dialog is confirmed, the technical instance split is done later in the roadmap step *Execution* in phase `ASCS_SPLIT`. The SUM creates the needed file system infrastructure in the IFS as the new directory `ASCS<nn>`. Even in this case, the former shadow instance number can be reused (see also chapter 4.4.2 *ABAP Upgrade Phase ASCS\_ASK* for more details). If the near Zero Downtime Maintenance option is used, the shadow instance number cannot be reused.

### 4.3.2 Shadow Instance File System

A shadow instance requires a similar file system structure as any other SAP instance. For both stack types, the SUM creates the needed files inside the SUM directory in the following top-level directories:

- ABAP: `/<path-to-SUM>/SUM/abap/system/<SID>`
- Java: `/<path-to-SUM>/SUM/sdt/system/<SID>`

These directories contain not all, but the typical parts of the SAP system directory structure. The instance specific directories and directory `SYS` (with subdirectories `global` and `profile`) can be found in the top-level directory. For an ABAP shadow instance, there is a `DVEBMGS<nn>` directory. A shadow ASCS instance is not used and therefore not created. Java shadow instances are always created in pairs, a Java server instance `JC<nn>` and its related central services instance `SCS<nn>`. In all cases, the instance directories contain a `work` directory, into which developer traces of the instance processes are written. Typically, the SAP kernel directory `exe` exists, but is empty. The SAP kernels used by shadow instances have a different location:

- ABAP: `/<path-to-SUM>/SUM/abap/exe`
- Java: `/<path-to-SUM>/SUM/sdt/data/kernel`

## 4.4 Important Phases for Technical Decisions

There are a few upgrade phases where important decisions are made by the user about how the SUM will work. These phases provide dialogs to ask for technical details needed for the procedure. They usually apply to upgrades for ABAP stacks only.

### 4.4.1 ABAP Upgrade Phase `INITSUBST`

SUM starts the *Configuration* roadmap step with phase `INITSUBST` and corresponding dialogs. In this phase, the first important decisions are made about how the upgrade procedure will run. The phase consists of up to two dialogs. The first dialog asks for the general upgrade setup (see [figure 21](#) on page 33), while the second dialog appears afterwards to specify detailed upgrade configuration parameters, like degree of parallelism or the batch host configuration.

The first major decision for the upgrade process is to choose the right upgrade strategy in terms of resource usage and consumption. This decision also has significant influence on the total runtime of

the upgrade procedure. A detailed description is available in the SUM upgrade guide in chapter *Pre-configuration Mode Planning*.

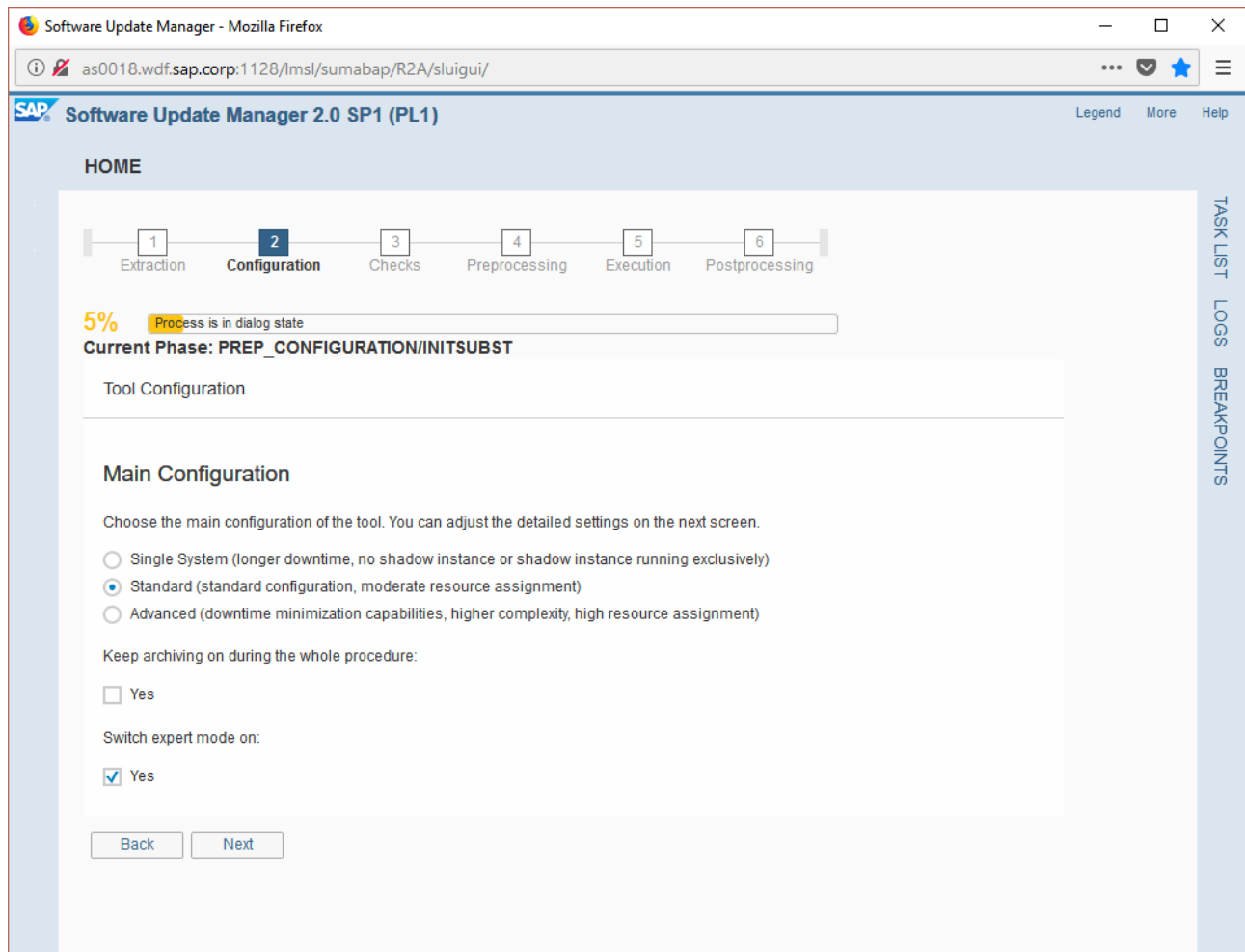


Figure 21: First dialog of phase INITSUBST

- In general, the *Standard* option will fit in almost all cases. The shadow instance will run in parallel to the original instance. Therefore, SAP users can stay active longer in the system. This mode requires enough hardware resources to run both instances at the same time.
- The option *Single System* ensures that the SUM procedure can run on machines with limited resources. This may be suitable for small test systems, or if the system to be upgraded runs in parallel to another SAP system and the performance degradation due to the running upgrade should be as low as possible. An upgrade with SUM always requires a certain amount of resources, but the detailed settings allow one to define a range with acceptable usage. In case a shadow instance is needed during the upgrade for technical reasons, the shadow instance runs only while the original instance is down and vice versa. Therefore, the setup of the shadow instance is also started much earlier than when using the *Standard* option.
- The *Advanced* option provides options for special requirements. The goal is to minimize the runtime for roadmap step *Execution* to a minimum. This makes the upgrade more complex and requires more hardware resources. The most prominent capability in that mode is the near-Zero Downtime Maintenance (nZDM), see chapter 6.2 *Near Zero Downtime Maintenance*.

Another important decision affects the journal handling during the *Execution* step of the upgrade procedure. The default setting is to turn journaling off for performance reasons. When this is done, the journal receivers will be switched from their standard name and the operating system takes the tasks to manage them. This happens in upgrade phase `DOWNCONF_TRANS`, right before the upgrade preparation phases have finished. From that point in time on, the journal settings are switched to a system-managed mode, in which a full journal receiver is automatically detached, replaced by a newly generated one and then deleted. With this mode being activated, housekeeping activities on full journal receivers can and should be paused because they are now being deleted automatically. It is essential to run all backups as requested by the SUM for this system-managed journal mode, otherwise there is no valid sync point for the SAP system and the upgrade tool to use in case a disaster recovery is ever needed. The journal will be switched back to the normal mode before roadmap step *Postprocessing* starts. If the default is changed and the journal is kept active for the whole SUM runtime, be aware that many database changes occur during downtime, causing the number of journal receivers to grow quickly, consuming several hundreds of GB disk space.

Enter the maximum number of ABAP processes (dialogue and batch) during the upgrade:

*ABAP PROCESSES (UPTIME)	3
*ABAP PROCESSES (DOWNTIME)	4

Enter the maximum number of parallel processes for execution of SQL commands:

*SQL PROCESSES (UPTIME)	3
*SQL PROCESSES (DOWNTIME)	6

Enter the maximum number of parallel import processes:

*R3TRANS PROCESSES (UPTIME)	3
*R3TRANS PROCESSES (DOWNTIME)	6

Enter the maximum number of parallel R3load or table comparison processes:

*R3LOAD PROCESSES (UPTIME)	8
*R3LOAD PROCESSES (DOWNTIME)	40

Figure 22: Dialog options for parallel jobs in phase `INITSUBST`

The second dialog of upgrade phase `INITSUBST` prepares the degree of parallelism to be used in certain upgrade phases. There are four options to be defined as shown in [figure 22](#). Each option consists of two values – one process limit for the time while the standard instance is up and running and another one while it is down, see also chapter 5.2 *Parallelism During the Upgrade*.

There are also other options available to optimize the overall upgrade process. Among them are options for transaction `SGEN` for the ABAP stack. Typically, when running a release upgrade, ABAP loads (precompiled ABAP sources) get invalidated due to changes in the kernel, the ABAP code and new DDIC or database structures. The rebuild of those loads is a time and resource consuming task and may take several hours, depending on the number of ABAP sources in the system. On the other hand, the response times will slow down after the upgrade if the loads are not prepared. In that case, any ABAP load gets automatically refreshed on first touch. This might be sufficient for test systems, but productive systems should be well prepared to be ready for use after the upgrade.

#### 4.4.2 ABAP Upgrade Phase ASCS\_ASK

If an upgrade runs with the SUM for an ABAP stack and the affected SAP system does not already have an ASCS instance (which is a directory called `ASCS<nn>` in directory `/usr/sap/<SID>`), a dialog appears to ask whether the missing instance should be created and, if confirmed, which instance number should be assigned. As default, the already known shadow instance number is proposed, but it can be overwritten. In general, it is a good choice to confirm both the creation of the instance and the reuse of the shadow instance number. If there are company internal rules to assign instance numbers, a different, previously unused number can be defined. If an undesired instance number was selected for the shadow instance by accident (for example by just confirming the proposal calculated by SUM), all ILE and IFS based parts of this instance get deleted at the end of the upgrade. After the cleanup of the SUM, the formerly used shadow instance number is considered again “unused” on that host. When reused as ASCS instance, the change of that number is still possible but requires a higher effort and an additional short downtime of the SAP instance to enable the modified system configuration. As usual, numbers above 90 must not be used.

The ASCS instance decouples certain services from the primary application server instance and provides the enqueue server and message server. This is also the new default setup when creating a system copy of an ABAP system using the SAP Software Provisioning Manager (SWPM). The real technical split of the instances will be done later at the end of the *Execution* roadmap step in upgrade phase `ASCS_SPLIT`. Besides the preparation of the file system structures in the IFS, the SUM also creates the instance profile for the new instance and adopts the affected values to the new system configuration.

### 4.5 Important Platform Specific Phases

The SUM is compiled for use on specific DB/OS combinations (such as Db2 for IBM i / IBM i), but the appearance and the workflow are identical for all combinations when running the same scenario. Most of the platform specific code runs transparently for the user. Nevertheless, there are some specific phases which run only for a given platform. Those upgrade phases can often be recognized from their names, having a prefix or suffix in the name indicating the specific use. For IBM i, the names OS4 (short for the former OS name OS/400) and/or DB4 (short for the former DB name DB2/400) are used within the SUM.

The most important platform specific phases are described in the next chapters. There are many more, which can easily be identified by their phase name prefix or suffix. Not all of them require specific actions, but simply run in a different way compared to other platforms due to the nature of different operating systems and database management systems.

#### 4.5.1 Platform Specific Phases for ABAP Stack Upgrades

Very early on in processing, SUM checks some configuration values of the source release of the given SAP system. These checks are consolidated in phase `PREP_INFRASTRUCTURE_OS4`. The SUM verifies the currently used user concept, release and patch level of the active SAP Kernel, the SAP Host Agent and more. This phase stops the SUM if requirements are not met. For every check that failed, the corresponding SAP Note is referred to, which points to the necessary actions. Currently, the following SAP Notes can be relevant:



- [SAP Note 1149318](#) User concept conversion using the tool CONVUSRCPT
- [SAP Note 1031096](#) Installing Package SAPHOSTAGENT
- [SAP Note 1907153](#) SAP JVM Switch Tool: FAQ

There are other relevant SAP Notes in this area, which are not mandatory for the SUM. If the SAP system is not yet configured that way as described in those SAP Notes, it will be adopted automatically during the upgrade procedure:

- [SAP Note 1078134](#) IBM i: Distribution of ILE and PASE system components
- [SAP Note 1632754](#) IBM i: Changeover to instance-specific directory

In this phase, the upgrade tool libraries get created from the file `ILE.SAR`, which is shipped with the SUM archive. For SUM version 1.0, the upgrade tools are based on SAP Kernel release 7.21, while SUM version 2.0 relies on SAP Kernel release 7.53<sup>4</sup>.

In some cases, the SUM uses very new features from the operating system or the database. Therefore, at the beginning of the roadmap step *Configuration*, phase `PTF_CHECK_OS4` verifies the existence of several PTFs, depending on the currently used version of the operating system, the upgrade target release and required features for certain upgrade scenarios. This PTF check is not identical to the one provided by SAP Kernel command `CHKR3PTF` and the underlying InfoAPAR file. In this phase, only PTFs required for the currently running upgrade will be verified, dependent on the planned target release and chosen upgrade strategy. Nevertheless, it is always recommended to execute the check using the InfoAPAR to keep the IBM i server up-to-date with latest SAP relevant fixes.

If you plan to upgrade both the IBM i OS and SAP applications, perform the OS upgrade first if the new OS level is allowable for the existing SAP applications. This will allow SUM to use as many new functionality and performance improvements as possible from a more current OS version.

The upgrade phase `DOWNCONF_TRANS` is not marked as a platform specific phase and runs on every supported DB/OS combination. Nevertheless, it executes very specific actions. This phase is one of the last phases before an upgrade enters the downtime module. For that reason, it requires to create a new backup of the SAP system including the complete SUM directory (if it is created outside of path `/usr/sap/<SID>`). This backup is important in case of a disaster recovery situation that may occur during the upgrade downtime. Once this backup has been restored for the SAP system, any decision about resetting or continuing the upgrade can be revised again. After the backup of the database has been created and the dialog has been confirmed, the journal gets switched into upgrade mode. This upgrade mode allows the OS to auto-delete any full journal receiver without saving it. This means those receivers cannot be used for recovery because not the complete data change history is available for that. The name of the journal receivers (objects of type `*JRNRCV`) change to `UPGRAD<nnnn>` indicating this upgrade mode. In contrast to that, the standard mode uses the name `QSQJRN<nnnn>`. The name of the journal itself (object of type `*JRN`) is kept. During the phase `INITSUBST`, the corresponding dialog allows to enable option “*Keep archiving on during the whole procedure*” to keep the normal journal mode. In that case, the user must take care of disk usage. Many database changes are processed during downtime causing quickly growing journal receivers. If that option was chosen, the database journal will not be switched and the user has to take care manually about sufficient disk resources to store all the journal receivers.

Phase `STARTSAP_PUPG` is the counterpart of upgrade phase `DOWNCONF_TRANS`. Here again, the SUM requests to perform a backup as the upgrade has now completed from a technical point of view and the SAP system has reached the target release. In this phase, the journal processing is switched

---

(4) Information based on March 2018



back to the normal mode. The status of the journal can be checked using following OS command. By pressing key F15, the command provides more details about attached journal receivers and their status.

```
WRKJRNA JRN (<DB-library>/QSQJRN)
```

```
AS0018
File Edit View Communication Actions Window Help
Work with Journal Attributes
Journal . . . . . : QSQJRN Library . . . . . : R3R2ADATA
Attached receiver . . : QSQJRN0003 Library . . . . . : R3R2AJRN
Text . . . . . : *BLANK
ASP . . . . . : 1
Message queue . . . . : QSYSOPR
Library . . . . . : *LIBL
Manage receivers . . . : *SYSTEM
Delete receivers . . . : *YES
Journal cache . . . . : *NO
Manage delay . . . . . : 10
Delete delay . . . . . : 10
Journal type . . . . . : *LOCAL
Journal state . . . . . : *ACTIVE
Minimize entry data : *FILE
Journalled objects:
Current . . . . . : 46694
Maximum . . . . . : 10000000
Recovery count . . . . : *SYSDFT
Receiver size options: *RMVINTENT
*MAXOPT2
Fixed length data . . : *JOB
*USR
*PGM
Bottom
F13=Display journaled files F14=Display journaled access paths
F15=Work with receiver directory F24=More keys
MA A 01/001
1902 - Session successfully started
```

Figure 23: Output of WRKJRNA command

The last action of the SUM runs phase CLEAN\_INFRASTRUCTURE\_OS4. This upgrade phase removes the ILE libraries created and used during the upgrade. For that, SUM starts the helper program `o4upgtools`. Actions are documented in logfile `o4upgtools_CLEANUPINFRA.LOG`. If this logfile indicates that libraries could not be deleted due to locking issues, it is safe to delete those libraries manually after the upgrade has been finished.

#### 4.5.2 Platform Specific Phases for Java Stack Upgrades

Also, for Java upgrades, the SUM begins with the source release checks for the given SAP system. These checks are executed in phase `OS4-PREREQ-CHECKS`. For historic reasons, the Java part of SUM runs the prerequisite checks and the preparation steps for the infrastructure in two different phases. Therefore, right after the prerequisite checks, phase `PREP-INFRASTRUCTURE-OS4` creates the infrastructure required for the SUM procedure. Since SUM version 2.0 cannot be used for Java upgrades, only the tools library based on the kernel release 7.21 is loaded.

Some years ago, an old version of the SAP installation tool `SAPinst` provided an incorrect format for the JDBC connection URL in the secure store, used for database connect from the Java stack. For

the standard use of the Java stack, the installed format did not cause problems and the connection to the database worked fine. But when starting an upgrade, the misleading URL format could cause data loss, in the worst case by deleting tables accidentally. Upgrade phase CHECK-SECURE-STORE-DB4 has been introduced for a check to avoid this situation of data loss. The SUM stops when it detects this bad URL format. Before repairing the secure store in the SAP system, the current SUM cannot continue and must be reset, as some calculations for the shadow database have already been done at this point in time. In addition, a system downtime is required for the repair to activate the changed URL. Therefore, the repair should be done before a planned downtime. The following SAP Note contains details:

- [SAP Note 1665188](#) IBM i: Upgrade causes data loss in productive DB

For the access to the Java databases SAP software uses Java programs combined with the JDBC driver. On IBM i servers, this is the file `jt400.jar`, which is part of the *IBM Toolbox for Java*. This toolbox is based on open source and can be downloaded from the [JTOpen project at Sourceforge](#). Further descriptions about the toolbox driver can be found in the IBM Knowledge Center article [The IBM Toolbox for Java JDBC driver](#). The phase DETERMINE-NEW-JDBC-DRIVER-LOCATION provides a dialog, asking for an updated version of the JDBC Toolbox Driver to be used for the SAP system after the upgrade. This file, called `jt400.jar`, is needed for database connections between IBM i for Db2 and Java programs. The version of this file cannot be derived from its name, but it is important when upgrading to products based on SAP NetWeaver 7.4 or higher. The version information can be retrieved by the following call, executed in a QP2TERM session. Java version 4 or higher can be used for this call, provided by any JVM vendor. The example below indicates compliance with JDBC version 4.0, required for SAP Java products based on SAP NetWeaver 7.4 or higher.

```
> java -cp /usr/sap/R1J/download/jt400.jar utilities.AboutToolbox

IBM Toolbox for Java:
Open Source Software, JTOpen 8.1, codebase 5770-SS1 V7R2M0.00 built=20131021 @J5
Supports JDBC version 4.0
Toolbox driver version 10.1
$
```

The SUM does not expect the latest kernel patch level to be active in the source release of the SAP system, but there are required minimum kernel patch levels. The upgrade phase OS4-MINIMUM-KERNEL-CHECK verifies a suitable kernel version to avoid know issues during the procedure. More details can be found in

- [SAP Note 2375098](#) IBM i: Kernel Requirements for Software Update Manager

Once SUM has finished the upgrade, it runs some cleanup tasks. The removal of the shadow instance and its shadow database schema is a time-consuming task. For Java upgrades, phase DROP-SHADOW-SCHEMA-OS4 starts one or two batch jobs for that via SBMJOB command right at the end of the SUM procedure. These jobs are always called UPGDLTSHDJ and run with user profile <SID>ADM. It is possible for large systems that those jobs are still running while SUM itself has already finished.

The last action of SUM runs phase CLEAN-INFRASTRUCTURE-OS4. This upgrade phase runs the same actions as for the ABAP upgrade. If the jobs from phase DROP-SHADOW-SCHEMA-OS4 run longer than the remaining SUM processes, the Java upgrade kernel library SAP<SID>UPGJ might be locked and cannot be deleted immediately. After the UPGDLTSHDJ jobs have finished, it is safe to delete this library manually.

## 5 Performance Considerations

### 5.1 Long Running Upgrade Phases

Running upgrades for SAP systems is a very time and resource intensive task and requires good preparation. Besides the pure SUM process, a machine check-up and a review of the installed SAP system landscape need to be taken into account. When started, the SUM generates a noticeable workload to the affected server or logical partition with impact on all installed applications (SAP and non-SAP). Experience shows that out of the several hundreds of upgrade phases, only 10 to 15 phases dominate the overall runtime of the SUM tool. The following example was taken from a complete landscape with test, quality assurance and productive SAP system, upgrading from release SAP ECC EHP4 (based on release SAP\_BASIS 701) all the way to SAP ECC EHP7 (based on SAP\_BASIS 740) and including about 25 add-ons by SAP and third parties:

- Database size: about 1.4 terabytes (for QA and productive systems, test system was smaller).
- One LPAR with just the productive SAP system.
- Another LPAR with the test system, the quality assurance system and an SAP Solution Manager in parallel.

In total, the SUM ran about 750 phases for the complete upgrade and took a total runtime of about 34 hours for the productive SAP system and more than 56 hours for the quality assurance system (without wait times during the dialog phases). Both LPARs were Power7 servers with almost the same hardware setup. The following overview shows the 20 most long running upgrade phases for both upgrades in descending order of their runtimes.

Quality Assurance System		
Roadmap step	Phase name	Runtime
Preprocessing	SHADOW_IMPORT_INC	09:24:43
Downtime	TABIM_UPG	04:50:18
Preprocessing	DDIC_UPG	03:47:54
Configuration	EHP_INCLUSION	01:21:29
Configuration	CONFLICT_CHECK	01:17:14
Downtime	PARCONV_UPG	00:51:30
Preprocessing	DBCLONE	00:29:14
Preprocessing	SCEXEC_TRIG	00:27:10
Preprocessing	TP_ACTION_CP2STAB	00:25:21
Preprocessing	RUN_RSAIMMERGE_UPG_SHD	00:22:47
Configuration	TR_CMDIMPORT_PREPARE	00:19:18
Preprocessing	TABIM_REDUCE	00:16:58
Preprocessing	TABIM_POST_SHD	00:16:08
Preprocessing	CMDFILE_UPG	00:15:46
Downtime	PARMVNT_XCNV	00:15:34
Preprocessing	JOB_RUTSETRCST	00:15:09
Preprocessing	PARDIST_SHD	00:14:41
Preprocessing	PARMVNT_SHD	00:13:39
Postprocessing	JOB_RASUVAR2	00:12:53
Downtime	XPRAS_AIMMRG	00:12:43

The upgrade of the productive system resulted in a different order, and the phases listed in both tables have partially significantly different runtimes (e.g. phase EHP\_INCLUSION). In both upgrades, only five phases consistently dominate the total runtime for the whole process, so it is worth considering these runtimes more in detail for possible optimizations.

Productive system		
Roadmap step	Phase name	Runtime
Preprocessing	SHADOW_IMPORT_INC	09:30:57
Configuration	EHP_INCLUSION	05:40:06
Downtime	TABIM_UPG	04:37:21
Preprocessing	DDIC_UPG	03:35:09
Configuration	CONFLICT_CHECK	01:14:07
Downtime	PARCONV_UPG	00:26:47
Postprocessing	JOB_RASUVAR2	00:21:45
Preprocessing	RUN_RSAIMMERGE_UPG_SHD	00:21:27
Preprocessing	SCEXEC_TRIG	00:20:58
Preprocessing	TABIM_REDUCE	00:17:22
Preprocessing	DBCLONE	00:16:36
Configuration	TR_CMDIMPORT_PREPARE	00:16:23
Postprocessing	SAVELOGS	00:16:02
Preprocessing	CMDFILE_UPG	00:14:57
Preprocessing	TABIM_POST_SHD	00:14:19
Preprocessing	PARDIST_SHD	00:13:58
Preprocessing	JOB_RUTSETRCST	00:12:43
Preprocessing	PARMVNT_SHD	00:12:42
Downtime	XPRAS_AIMMRG	00:09:18
Postprocessing	CREATE_UPGEVAL	00:09:07

The phase descriptions in the next chapters are based on common experience and provide a short overview regarding the most expensive phases from the examples above. Depending on the specific SAP system and used upgrade scenario, other phases may also show long runtimes. The SAP Wiki page [SUM Phases](#) provides additional phase descriptions for the SUM. The phases in the next paragraphs are shown in the order of their processing sequence during the upgrade.

### 5.1.1 ABAP Phase EHP\_INCLUSION

Logfile	Comment
EHP_INCLUSION.LOG	Standard upgrade tool logfile for the phase
SPDA_EHP_INCLUSION.LOG	Logfile for ABAP work processes used during the phase
EHP_INCLUSION_PROCS.*	OS environment and command lines for OS used processes

This upgrade phase runs in roadmap step *Configuration*. The goal is to extract, verify and upload all ABAP relevant packages from the upgrade download stack into the EPS inbox of the transport directory and to calculate the queue on how add-ons, enhancement packages and support packages will be applied. SUM triggers batch work processes in the ABAP stack via RFC to process the requests. By default, three additional OS processes will be started for the necessary database tasks. Depending on

the number of packages, this may take some time. In the examples above, more than 300 ABAP packages were processed during this phase, causing the observed runtime. The number of the parallel OS processes can be adjusted using the upgrade parameter file `SAPup_add.par`, but you must keep in mind that this phase is running during the uptime of the original SAP instance. Especially when the OS processes run the programs `tp` or `R3trans`, a high CPU workload may occur and have considerable effects on the dialog response times if the number of processes was chosen too high.

### 5.1.2 ABAP Phase CONFLICT\_CHECK

Logfile	Comment
CONFLCHK.LOG	Standard upgrade tool logfile for the phase
SPDA_CONFLCHK.LOG	Logfile for ABAP work processes used during the phase

In complex SAP systems, many components and add-ons have to fit together. In general, the SAP Solution Manager calculates valid stacks that fulfill all prerequisites for the upgrade process, but SUM also verifies the completeness and consistency of the provided ABAP packages. The number of packages to be checked clearly affects the phase runtime. There are no options to speed up this serial process. The phase is part of roadmap step *Configuration*.

### 5.1.3 ABAP Phase DDIC\_UPG

Logfile	Comment
PHASES.LOG	Standard upgrade tool logfile for the phase
SAPH<transport-id>.<SID> YI3H<transport-id>.<SID>	Many different logfiles with names known from the transport directory, at least one for each processed package. The relevant logfiles should be checked using the phase timestamps.

During phase `DDIC_UPG`, the data dictionary (DDIC) relevant parts in SAP packages and changes in ABAP dictionary objects are imported. Dictionary data describe the SAP view of the data in the underlying database. Affected objects are ABAP domains, data elements and tables. The phase applies to SAP support packages of the target release provided with the upgrade target stack to the shadow instance and modifies the dictionary view to the data. The work is distributed to parallel `R3trans` processes. In such transport alike phases, the logfile `SAPupConsole.log` shows the progress and the degree of parallelism (see [figure 22](#) on page 34) while working on different prepared data packages. In the given example, 23 packages were processed in total with an overall size of more than 21 GB (cumulated size of the data files of all mentioned SAP packages).

```
>> 2017/10/31 16:54:17 START OF PHASE MAIN_SHDRUN/DDIC_UPG

Running /usr/sap/PRD/SUM/abap/exe/tp pf=/usr/sap/PRD/SUM/abap/var/SHADOW.TPP checkimpdp PRD

0:00: Importing for group 1/23 size 2158.78 MB: Transports SAPK-740BRINSAPBASIS-SAPK-731BMINSAPBW.
Running /usr/sap/PRD/SUM/abap/exe/tp pf=/usr/sap/PRD/SUM/abap/var/DDICUPG.TPP put PRD -Dmainimp_proc=1
-Dparallel=10

26:59: Importing for group 2/23 size 296.70 MB: Transports SAPK-740BOINSAPABA-SAPK-740BMINSAPBASIS.
Running /usr/sap/PRD/SUM/abap/exe/tp pf=/usr/sap/PRD/SUM/abap/var/DDICUPG.TPP put PRD -Dmainimp_proc=2
-Dparallel=5

30:16: Importing for group 3/23 size 4238.61 MB: Transports SAPK-617BOINMDGAPPL-SAPK-617BMINSAPAPPL.
Running /usr/sap/PRD/SUM/abap/exe/tp pf=/usr/sap/PRD/SUM/abap/var/DDICUPG.TPP put PRD -Dmainimp_proc=1
-Dparallel=10
. . .
```

### 5.1.4 ABAP Phase SHADOW\_IMPORT\_INC

Logfile	Comment
PHASES.LOG	Standard upgrade tool logfile for the phase
SAPK<transport-id>.<SID> YI3K<transport-id>.<SID>	Many different logfiles with names known from the transport directory, at least one for each processed package. The relevant logfiles should be checked using the phase timestamps.

The upgrade phase SHADOW\_IMPORT\_INC is the counterpart of phase DDIC\_UPG in terms of data content vs. data structures. Parallel R3trans processes fill shadow tables with new data from add-ons and support packages to get them ready for use with the target release. The progress on processing data packages is shown in SAPupConsole.log. In the given customer example, 40 packages were calculated with a total size of about 29.4 GB. Note here the marked 4<sup>th</sup> character in the logfile names that differentiate the logfiles from different phases working on the same SAP data packages.

### 5.1.5 ABAP Phase TABIM\_UPG

Logfile	Comment
PHASES.LOG	Standard upgrade tool logfile for the phase
SAPI<transport-id>.<SID> YI3I<transport-id>.<SID>	Many different logfiles with names known from the transport directory, at least one for each processed package. The relevant logfiles should be checked using the phase timestamps.

The upgrade phase TABIM\_UPG is part of roadmap step *Execution*. The SAP system is offline at this point of time, and the upgrade imports data into SAP internal control tables using parallel R3trans processes. In the example, this remaining data part was imported in 33 packages with a total size of about 9.5 GB.

### 5.1.6 Other Relevant Phases

The ABAP upgrade phases EU\_IMPORT1 and EU\_IMPORT2 run in roadmap step *Preprocessing* and import data into the shadow tables needed to run the shadow instance. These phases run program R3load to fill the tables with data. The phases are active in full release upgrades only and require an “Upgrade Export” media set being supplied. SAP provides such media sets as ISO images for DVDs or Blu-ray disks. These media sets do not need to be supplied as real physical media, if the image files can be mounted to the file system.

ABAP upgrade phase DBCLONE is the counterpart of the EU\_IMPORT phases for shadow tables that will be copied from their original database equivalents. In this phase, the data does not need to be imported from an upgrade export media, but the SAP system can provide it by itself. It is used to derive the minimum subset of tables needed by the SAP shadow instance from the original database. The main work of that phase is executed by ABAP work processes of the original instance.

Similar to the ABAP world, the Java deployment phases load data into tables of the Java database and ensure the synchronization between the new database content with their corresponding file system locations. The process is driven by running Java processes started by the SUM tool. The upgrade



distinguishes between shadow and standard deployments (defined by the database to be updated) and between offline and online deployments (decided by the run status of the affected Java instance). The overall runtime of such upgrade phases depends on the number of files to be processed. Java deployment phases are typically the longest running phases in Java upgrades, but in general they do not take as long as comparable ABAP upgrade phases due to the different sizes of the Java and ABAP databases. For example, when running an upgrade for a dual-stack SAP system, the total SUM runtime is always dominated by the ABAP upgrade part.

### 5.1.7 Long Running Phases in Java Upgrades

In this customer example, no Java stack was involved in the analyzed upgrade scenario. Nevertheless, also Java upgrades have phases with a significant longer phase runtime. But in general, Java upgrades run faster compared to ABAP upgrades.

The Java specific upgrade phase `IMPORT-JLOAD` is the counterpart to the ABAP `EU_IMPORT` phases and loads initial data into the Java database tables. It will be executed for so-called “switch-based” upgrades. In contrast to the “deploy-based” upgrades, shadow instances (a Java shadow instance and a shadow SCS instance) are created to run database changes before the downtime.

Phase `DEPLOY-SHADOW-INSTANCE` is the counterpart of ABAP phase `SHADOW_IMPORT_INC`. During this phase Java content is loaded based on the SAP Support Package stack included in the upgrade procedure. It requires the shadow instances to be up and running to verify the Java content. The Java shadow instance itself will be restarted several times automatically during phase processing.

## 5.2 Parallelism During the Upgrade

### 5.2.1 Upgrade Settings

The upgrade process itself is a serial process. A defined list of roadmap steps with their related phase lists provides a sequence to run several hundreds of upgrade phases. This sequence ensures the consistency of the SUM process and is basically the same for every upgrade. Deviations are derived from the SAP system configuration, such as the actual SAP product, the involved source and target releases, or the types and number of installed add-ons. Whenever possible, the SUM tries to use parallel processes within a given step, if the workload can be split and the user had chosen the appropriate options. The following SAP executables are typically involved in such cases:

Program	Comment	Upgrade phase examples
<code>disp+work</code>	ABAP work processes configured for batch workload to run ABAP reports in the original instance as well as in the shadow instance	<code>DBCLONE</code> , <code>ACT_UPG</code> , <code>PARDIST</code> or <code>XPRA</code> phases
<code>R3trans</code>	Process of transport related tasks, import of support packages, user and modification transports	<code>DDIC_UPG</code> , <code>SHADOW_IMPORT_INC</code> or <code>TABIM_UPG</code>
<code>R3load</code>	Import of data into tables	<code>EU_IMPORT</code> phases
<code>SAPuptool</code>	Running SQL statements against tables	<code>MVNTAB</code> , <code>PARCONV</code> or <code>SCEXEC</code> phases

The following SAP Note provides a detailed view on how parallelism is used with SUM, explains the dialog settings for upgrade phase `INITSUBST` and takes special capabilities of parallelism in program `R3trans` into account:

- [SAP Note 1616401](#) Parallelism in the Upgrade, EhPs and Support Packages Implementations

### 5.2.2 Hardware Considerations

When talking about upgrade performance, a major performance factor are the hardware capabilities of the system on which the upgrade is running. Looking closer at long running upgrade phases, it can be observed that jobs running in those phases are spending a lot more time doing I/O than CPU work. Typically, these jobs are moving data from the file system into the database or perform modifications on database tables, which require to create temporary copies of those tables. And since both the database and the file system are residing on the disk storage, the I/O subsystem is performing most of the workload.

In many cases, an SAP release upgrade goes along with a hardware upgrade, because newer SAP releases usually demand more hardware resources. The [IBM Insight for SAP](#) tool, available free of charge from the [IBM Techdocs](#) website, provides an analysis of the SAP system workload, which enables IBM to estimate hardware resource demand for the upgraded system.

If there are plans to upgrade the hardware before upgrading the SAP release, the new machine should be sized not only for sufficient CPU power, but also for I/O performance. This touches the type of I/O adapters being installed as well as the size, type, speed and number of disks that are placed into the storage subsystem. An optimally balanced and tuned I/O subsystem can make a difference of several hours in total upgrade runtime and downtime in particular. Removing hot spots from the storage subsystem is one example of tuning the I/O subsystem.

Solid State Disk (SSD) drives and Flash Storage Systems are providing an increased read performance compared to traditional spinning disk drives. Although the write performance of these storage types is also faster, a well-balanced spinning disk storage subsystem with caching and a sufficient number of disk arms can provide similar write performance. That is why mainly read-intensive tasks benefit from SSD and Flash Storage. With the integrated storage management of IBM i, the presence of fast drives in the disk pool alone results in an increased read performance. The higher the percentage of fast drives versus spinning drives in the pool is, the higher the read performance increase is. IBM i also provides the possibility to explicitly place objects on the faster disks. For physical files (objects of type `*FILE-PF`), this can be done with the OS command:

```
CHGPF FILE(<library>/<file>) UNIT(*SSD)
```

For SQL tables, the same can be accomplished with the SQL statement:

```
ALTER TABLE <library>/<table> ALTER UNIT SSD
```

The current SUM versions do not provide capabilities to keep this attribute if a table gets replaced by the upgrade. For new tables, this decision has not been made yet, so they are created on any available disk unit.

The white paper [SAP on IBM i Solid State Disk \(SSD\) Usage Recommendations](#) at the [IBM Techdocs](#) website has more details about this topic. In addition, the [IBM Knowledge Center for IBM i](#) contains further documentation about the tools available to monitor and tune system performance. Of course, improved I/O subsystem performance does not only affect the upgrade but also the runtime of the upgraded system.

### 5.2.3 Parallel Considerations for Programs `tp` and `R3trans`

The number of parallel processes to be used for certain upgrade phases can be defined by the user in phase `INITSUBST`. Additionally, some of the SAP programs used within the upgrade process will automatically split work packages on their own to process the pieces in parallel. The most prominent example of this is the program combination of `tp` and `R3trans`. The most visible indicator of this behavior are the parameters

```
-Dmainimp_proc=<n> and -Dparallel=<n>
```

shown in logfile `SAPupConsole.log`, where `<n>` is the number of parallel running `R3trans` processes launched by the controlling `tp` job. Typically, they are used in the phases `DDIC_UPG`, `SHADOW_IMPORT_INC` and the `TABIM_*` phases to process SAP transport files. The actual number of SAP transport files depends on the upgrade scenario and can reach several hundreds of files, so running in parallel can be very effective.

Altering a table structure in the Db2 for IBM i database is an I/O intensive task and therefore not so much limited by the number of available CPUs but rather by the performance of the I/O subsystem. Even with only two CPU cores available, it can make sense to use 4 or even up to 8 parallel processes to speed up such upgrade phases. This type of database tasks is usually handled by program `tp`. But even with a high performance I/O subsystem and plenty of CPU cores available, choosing the number of parallel processes too high may have a contrary effect as the parallelism may run into lock wait timeouts. Lock situations may appear for example if two tables are interconnected through a spanning SQL view over fields of both tables and both tables get altered simultaneously. The first job touching such a structure creates temporary locks on all dependent objects. Therefore, a second job cannot get the lock for the view until the first job has finished its task and will fail. Those failed `ALTER TABLE` operations have to be repeated sequentially after all other parallel alters have finished. Too many processes in parallel will increase the probability for the described race condition, possibly extending instead of shortening the total runtime of the alter table phase.

The SAP transport tool `R3trans` offers a feature that allows importing a single package with parallel processes. This feature can speed up all import related tasks during the upgrade downtime and also after the upgrade for standard SAP transports. Refer to chapter 5.2.5 *Parallelism for Modification Imports* for the needed configuration of this feature in transaction `STMS`. As imports are a more I/O than CPU intensive tasks, the number of parallel processes to choose depends on the I/O performance of your system. Splitting up the transport packages into several pieces produces some overhead, and there is also a limitation of how many pieces can be generated, depending on the package size. That is why we recommend setting the number of parallel `R3trans` processes to a maximum of eight for each individual package. More information can be found in

- [SAP Note 1127194](#) R3trans import with parallel processes

### 5.2.4 Archiving and Reorganizing Database Tables

Besides the improvements of the upgrade tools and SAP kernel programs, there are also different options to prepare the SAP system for an optimized upgrade run time long before you start the actual upgrade. One of them is archiving of data which is no longer needed for the daily business and, if needed, the reorganization of large database tables containing a high amount of deleted records.

Tables with Deleted Rows at 18.04.2018 11:28:00

Table Name	Rows	Deleted Rows	% Deleted	Row Length	Waisted Space (KB)
"/BI53021"	570.500	1.719.510	75	58	100.055
TST01	460.849	1.101.217	70	384	415.906
DB4TA50000	17.445.539	443.460	2	416	180.744
SQ3007_DB4	2.401	222.936	98	4.070	505.543
DB4IN50000	8.548.578	194.685	2	392	74.791
SNAP	852	167.843	99	3.334	556.415
SQ3001_DB4	2.978	161.798	98	4.718	434.498
RSEUINC	6.693	156.199	95	160	25.602
SQ3000_DB4	795	101.274	99	4.076	219.100
SQ1000_DB4	1.427	94.887	98	1.324	123.266
DDXTF	55	87.270	99	457	41.206
SQTEXT_DB4	4.080	85.184	95	16.274	147.140
SQHVAR_DB4	1.506	84.951	98	2.274	56.871
DB6AUDITDT	0	46.813	100	574	26.676
DD03L	1.079.107	34.536	3	600	20.330
TSP0C	281	22.901	98	678	17.248
WDR_A50004	13.355	20.179	60	268	350.498
DDLOG	9	16.805	99	144	27.901
EUIMPORT	0	16.766	100	338	6.304
DB6AUDITHD	0	15.603	100	878	14.432
SQ3003_DB4	269	13.372	98	880	12.277
SCPRVALS	159.557	12.010	7	722	8.753
SMIMCONT1	111.966	8.808	7	3.048	17.816
SQ3002_DB4	44	8.785	99	5.286	30.164
RSDDS50015	3.790	8.461	69	378	3.240
WDR_M50000	13.712	7.994	36	144	54.476

Figure 24: Deleted Rows Analysis in Transaction DB02

Reducing the database size by archiving application data as recommended by SAP data management guidelines is one option. Data is moved from the database tables to an archive, for which updates are no longer allowed from a business perspective. The data is still accessible as read-only from the application, but no longer occupies space in the table which is used by the application for daily operation. Both, the runtime performance of the SAP system after the upgrade (due to reduced data volume) and the upgrade runtimes itself can be improved (in case of table modifications using the ALTER TABLE SQL statements). Therefore, it is highly recommended to archive application data where possible shortly before the actual upgrade starts. Some prominent SAP Notes about archiving are:

- [SAP Note 178487](#) High increase of table BSIS, RFBLG
- [SAP Note 203545](#) Archiv. Profit center: Analysis of table GLPCA
- [SAP Note 1257133](#) CD: Archiving options for CDCLS
- [SAP Note 1572522](#) EDI/IDoc: Deleting and reorganizing IDocs with SARA

Another way to reduce database table sizes is to reorganize them. During system lifetime activities, database tables that are updated become fragmented over time. This can happen if the space occupied by deleted table records are not replaced immediately by new entries. Such situations can appear after restructuring the SAP system, for example when deleting an SAP client. Archiving application data

as mentioned above may result in a lot of deleted rows in certain tables, making them candidates for reorganization. Transaction DB02 can display the number of deleted records for the SAP database tables ([figure 24](#) on page 46). The reorganization of such a database table may take several hours depending on the table size, number of deleted records and system workload. In general, you should be able to use online reorganization, meaning the tables are reorganized while the SAP system is running. The reorganization will have an impact on the system response times, so it's recommended to perform the reorganization during a time with low workload on the SAP system.

Depending on the OS release level, certain limitations apply to online reorganization. Usage notes about running command RGZPFM online can be found on the IBM Support page [RGZPFM - Basics on Reorganize While Active](#). An offline reorganization is also possible but requires stopping the SAP system. Further details about reorganizing of SAP database tables and recommendations are listed in

- [SAP Note 84081](#) IBM i: Reorganization of database tables (RGZPFM)

### 5.2.5 Parallelism for Modification Imports

After the upgrade downtime phases have finished, the shadow tables are switched to become active in the upgraded SAP system. Before the system can be productive again, there are a few more tasks to complete. In case of modifications to standard SAP objects, the SUM shows in the upgrade preparation phases a list of affected objects that have to be adjusted for the new release. These adjusted modifications, developed and tested on a separate development system before the upgrade, must be transported into the upgraded SAP system. Depending on the number and size of the modifications, the imports may run for some hours. These imports are handled by the SAP programs `tp` and `R3trans`. Both programs can run with a parallel import feature if the Transport Management System (TMS) is configured accordingly.

The modification data is loaded with an import queue in transaction STMS and the parallel feature is enabled in the transport configuration of the transport domain controller. The configuration can be edited only on the domain controller system in client 000 with sufficient administrator authorization in transaction STMS. The option *Systems* from menu *Overview* provides the list of all connected SAP systems for the current transport landscape. The parallelism can be enabled per single SAP system or globally for all systems.

- Double click to the SAP system to be changed from the systems overview
- Switch to the tab *Transport Tool* and enable editing
- Insert a new line in the parameter list with value `parallel` in the *Parameters* column and a numeric value in the *Value* column as shown in [figure 25](#).

Refer to chapter 5.2.3 *Parallel Considerations for Programs tp and R3trans* section for recommendations how to choose the value. This setting can be kept from thereon to speed up all future transport tasks as well.

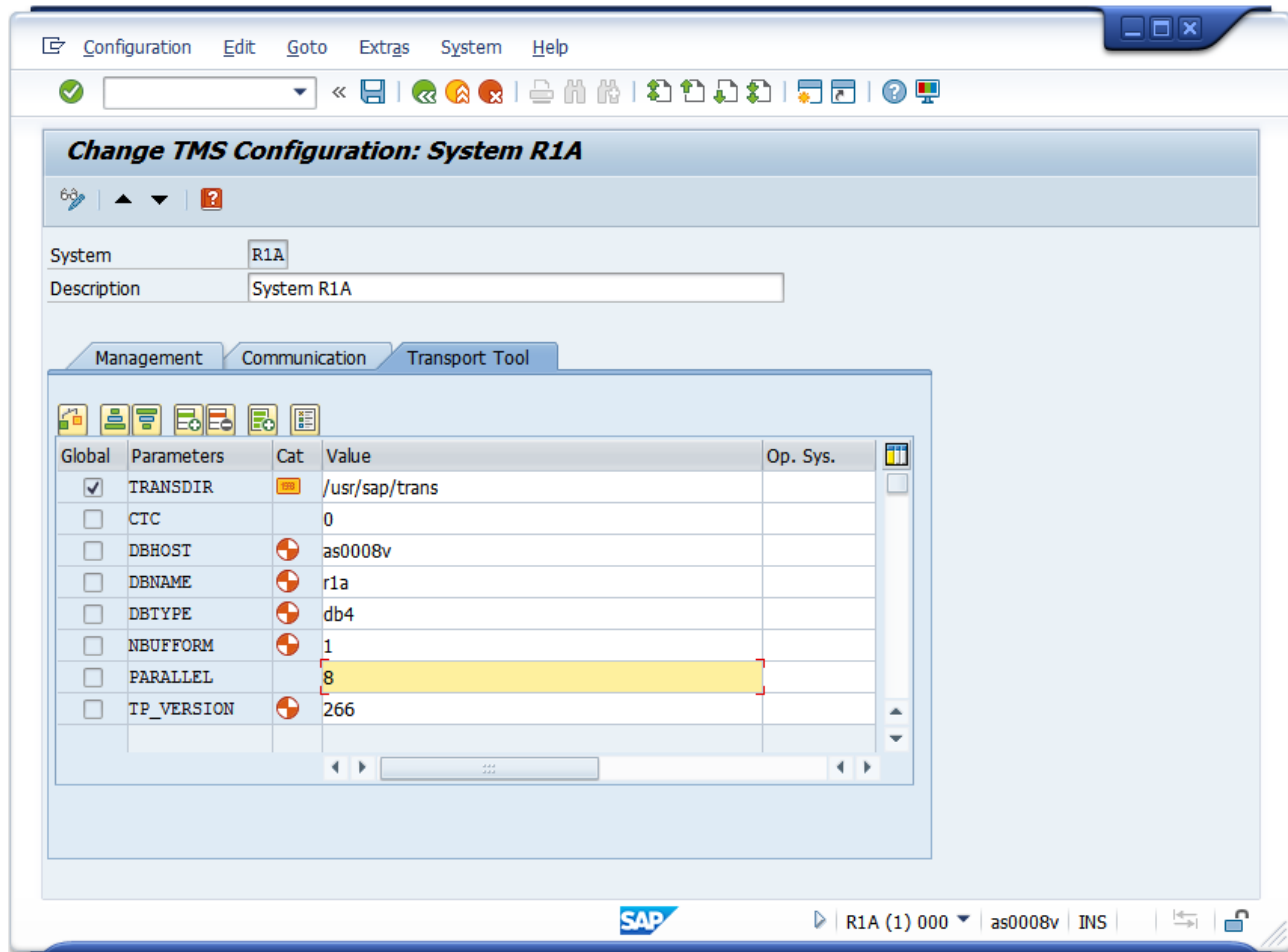


Figure 25: Enable Parallelism in the Transport Management System

### 5.3 SUM Performance Checklist

The following checklist summarizes actions in the chronological order of the upgrade process that should be taken in order to run the SUM upgrade procedure with best performance.

Before starting the SUM upgrade process:

- *Optional:* Running the [IBM Insight for SAP](#) tool to analyze current system workload. Get a recommendation for and implement hardware upgrades from IBM.
- Align the PTF level on your IBM i server with the PTF level listed in the current InfoAPAR for SAP on IBM i. Move to the latest IBM i release level compatible with your source SAP application.
  - [III4536 - SAP SUPPORT REQUIRED PTF LIST FOR IBM i 7.1](#)
  - [III4741 - SAP SUPPORT REQUIRED PTF LIST FOR IBM I 7.2](#)
  - [III4816 - SAP SUPPORT REQUIRED PTF LIST FOR IBM I 7.3](#)
- Check the storage subsystem balancing with OS command WRKDSKSTS to identify possible hot spots in disk usage and remove them if necessary ([SAP Note 517515](#)).
- Archive data as recommended by [SAP data management guidelines](#). This needs to be done before reorganizing tables.



- In case of large tables containing many deleted rows (ratio of deleted rows above 10%, e.g. after archiving), the reorganization of database tables with OS command `RGZPFM` might be helpful. Transaction `DB02` can be used to find applicable tables. A system downtime for certain tables might be required ([SAP Note 84081](#)).

During SUM roadmap step *Configuration*:

- Choose appropriate values for the number of parallel processes during SUM runtime in phase `INITSUBST`. The default values always fit, but might be suboptimal.

During SUM roadmap step *Postprocessing* (or after finishing the SUM procedure):

- Adopt the transport tool configuration in transaction `STMS` on the transport domain controller system for modification adjustment transports by adding the value `parallel`. A maximum value of 8 should be chosen on machines with a slow I/O subsystem.

## 6 Special Upgrade Strategies

### 6.1 Combined Upgrade and Unicode Conversion

With the release of SAP NetWeaver 7.5, SAP decided to support only Unicode codepages for products based on this SAP NetWeaver release or higher. Any SAP product based on component SAP\_BASIS 7.50 or higher cannot run in any ASCII codepage. Common examples in the SAP portfolio are SAP ECC 6.0 EHP8 or SAP BW 7.5. This causes some conditions to be considered when planning upgrade projects:

- SAP systems to be upgraded to releases based on component SAP\_BASIS 7.50 or higher must already run with Unicode codepage before the upgrade project starts.
- Therefore, SUM version 1.0 is the only SUM version that supports the combination of the upgrade process with preparation steps for a Unicode conversion afterwards.

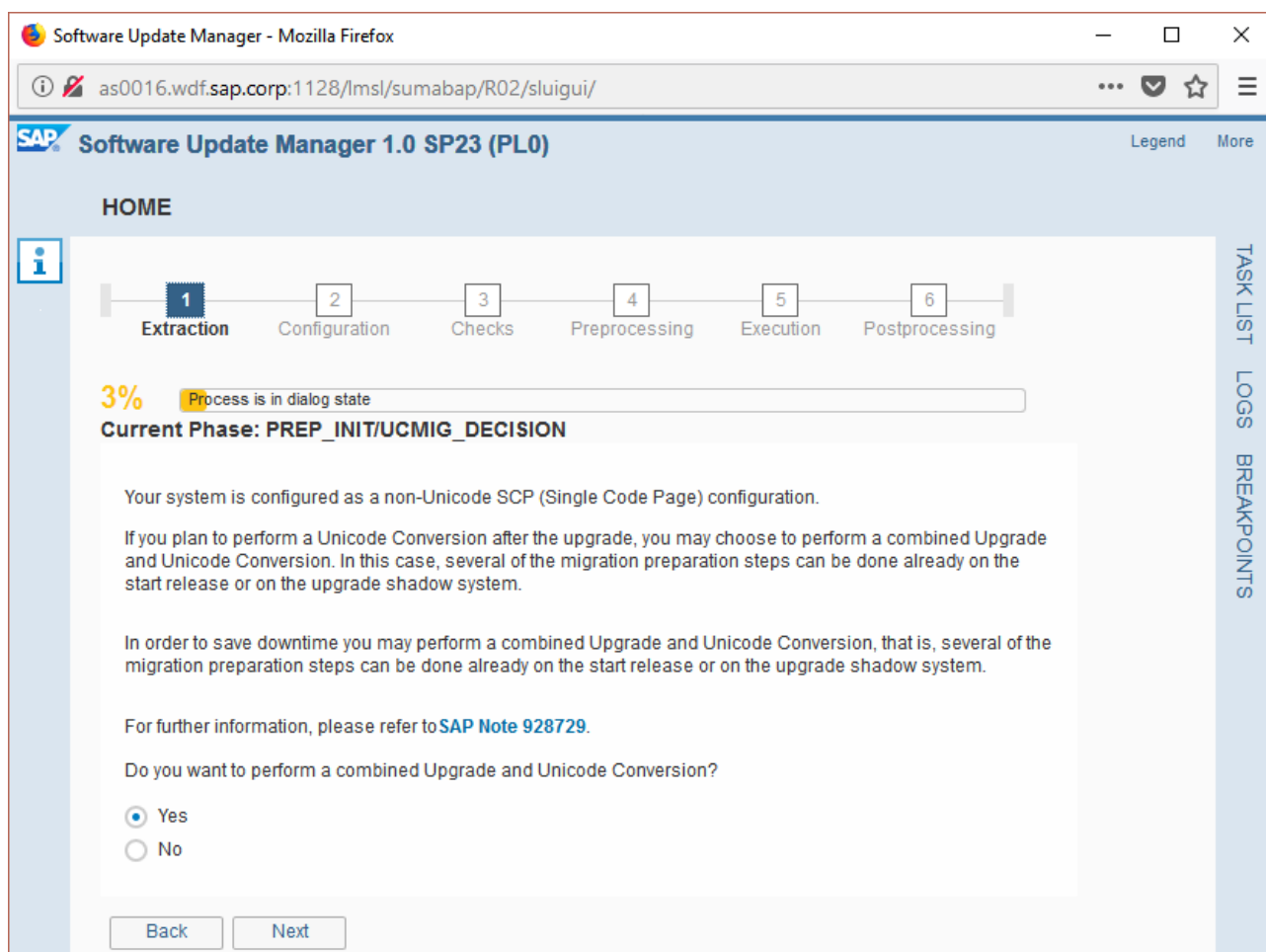


Figure 26: Decide an upcoming Unicode conversion during SUM

For upgrades to releases up to 7.40, there is an option to combine the upgrade with a Unicode conversion and let the SUM prepare certain tasks for a Unicode conversion right at the end of the upgrade. Upgrade phase UCMIG\_DECISION shows the corresponding dialog (figure 26). If the decision is “Yes”, the SUM prepares tables and data needed for a later conversion and runs checks to monitor the progress of the needed preparation steps for the Unicode conversion (e.g. running transaction SPUMG)

in phase UCMIG\_STATUS\_CHK. The tool suspends the upgrade process if needed. Also, other upgrade phases influence both process parts, for example phase RUN\_RADCUCNT\_EHPALL, where the Unicode nametab tables DDNTF\_CONV\_UC and DDNTT\_CONV\_UC are written during runtime of the shadow instance. Those tables get updated again in phase RUN\_RADCUCNT\_NEW during roadmap step *Execution* to reflect latest upgrade changes for DDIC objects.

It is very important to have all needed support packages for the target release included in the SUM procedure. The check point for that is upgrade phase BIND\_PATCH. Implementing any support package after the upgrade has finished but before the Unicode conversion starts would invalidate the Unicode preparation steps. In such cases, the preparations steps must be repeated completely. For the Unicode conversion itself, the tool SAP Software Provisioning Manager (SWPM) is used. The following SAP Notes provide additional documentation regarding the combination of the SUM upgrade with a Unicode conversion for an SAP system:

- [SAP Note 928729](#) Combined Upgrade & Unicode Conversion (CU&UC)
- [SAP Note 1322715](#) Unicode FAQs

In general, the whole database will be exported and imported again during a Unicode conversion. This is done by program R3load and driven by the SWPM tool. Depending on the size of the database, this can be a time-consuming task. For IBM i, another optimization can be used if the SAP system runs with a single **Latin-1** codepage (this is with entry 1100 in table TCPDB). In this case, the export affects only pool and cluster tables and not the complete database, which makes the procedure much faster. This optimized Unicode conversion is described in

- [SAP Note 800791](#) IBM i: In-place Code Page Conversion ASCII to Unicode

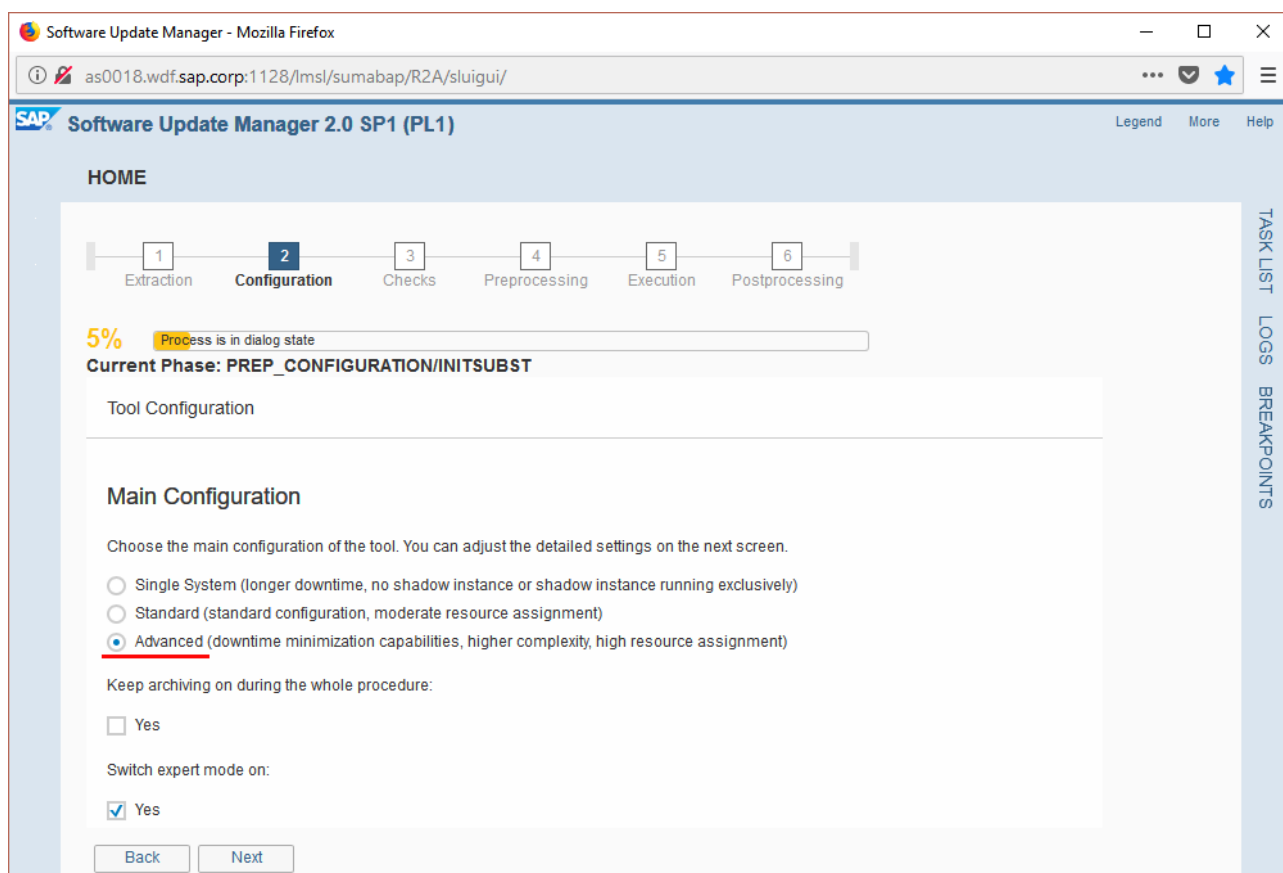


Figure 27: Enable downtime minimization methods in SUM

## 6.2 Near Zero Downtime Maintenance

An important amount of time during the upgrade is used for roadmap step *Execution*. At the beginning of this roadmap step, the SAP system will be shut down for the duration of this step. This is also called technical downtime. No SAP user can use the SAP system in this time, regardless of temporary instance starts in between. Phase `STARTSAP_PUPG` marks the end of this technical downtime. SAP spent a high effort to introduce techniques in order to optimize the SUM runtime in general and to shorten the time frame for roadmap step *Execution* in particular. One of these techniques is the approach of the near Zero Downtime Maintenance (nZDM). Central starting point for documentation is the upgrade guide in chapters *near-Zero Downtime Maintenance* and *Using the Record & Replay Technique in nZDM*, as well as following SAP Notes:

- [SAP Note 1678565](#) Prerequisites, Terms and Conditions for nZDM/SUM
- [SAP Note 1678564](#) Restrictions, Database-specific Settings and Troubleshooting of nZDM for SUM
- [SAP Note 1809339](#) IBM i: Support by nZDM/SUM for IBM i

To enable the nZDM feature in SUM, there is no special patch or additional software needed. The software package `SUM.SAR` contains all parts required for this method. It might be necessary for certain kernel source releases to implement minimum patch levels for SAP programs or to define profile parameters. In such cases, a short maintenance downtime (at least to restart the SAP instances) is required before the upgrade. To activate the feature, option *Advanced* needs to be selected in phase `INITSUBST` ([figure 27](#) on page 51).

In the following dialog, the checkbox needs to be marked to turn on nZDM support in the SUM as shown in [figure 28](#). The implementation of the nZDM methods contain database specific code, but the general procedure is the same for all supported platforms.

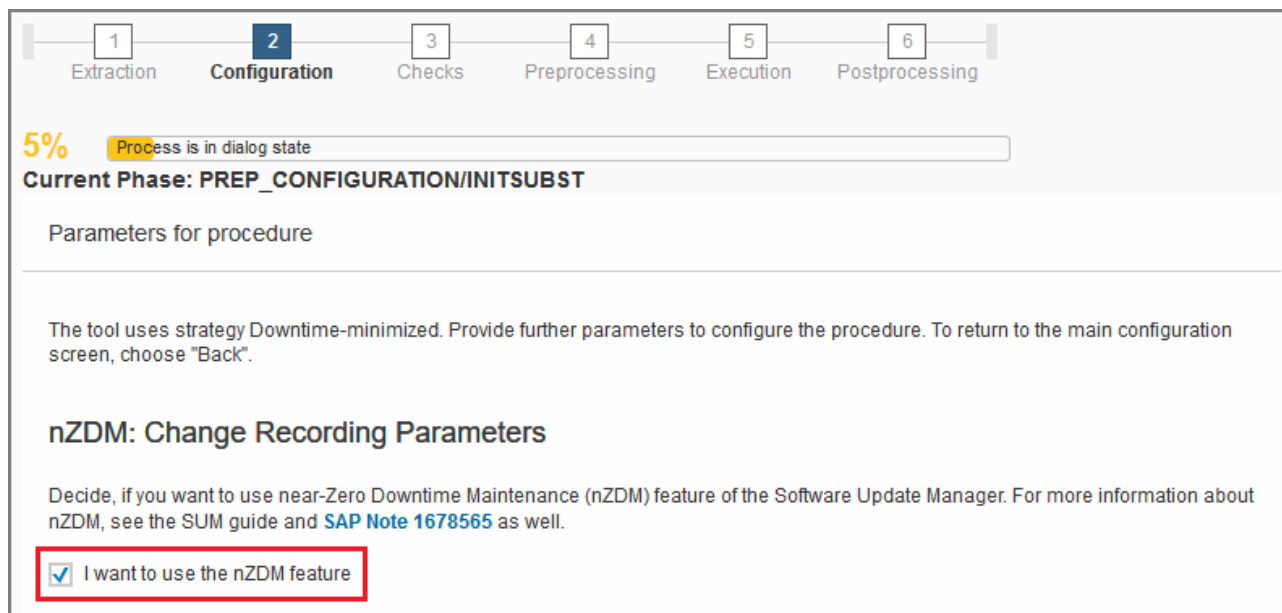


Figure 28: Turn on near Zero Downtime Maintenance

The goal of the nZDM technology is to reduce the technical downtime of the SAP system during an upgrade process, but the amount of needed actions and system modifications for the upgrade itself does not decrease. The process gets even more complex due to additional technical overhead to manage the change recording and replay. The SUM checks the progress of this data replication at the end

of roadmap step *Preprocessing* in phase `DOWNCONF_DTTRANS`. The upgrade downtime can be entered only after at least **95%** of all pending data replication has been done.

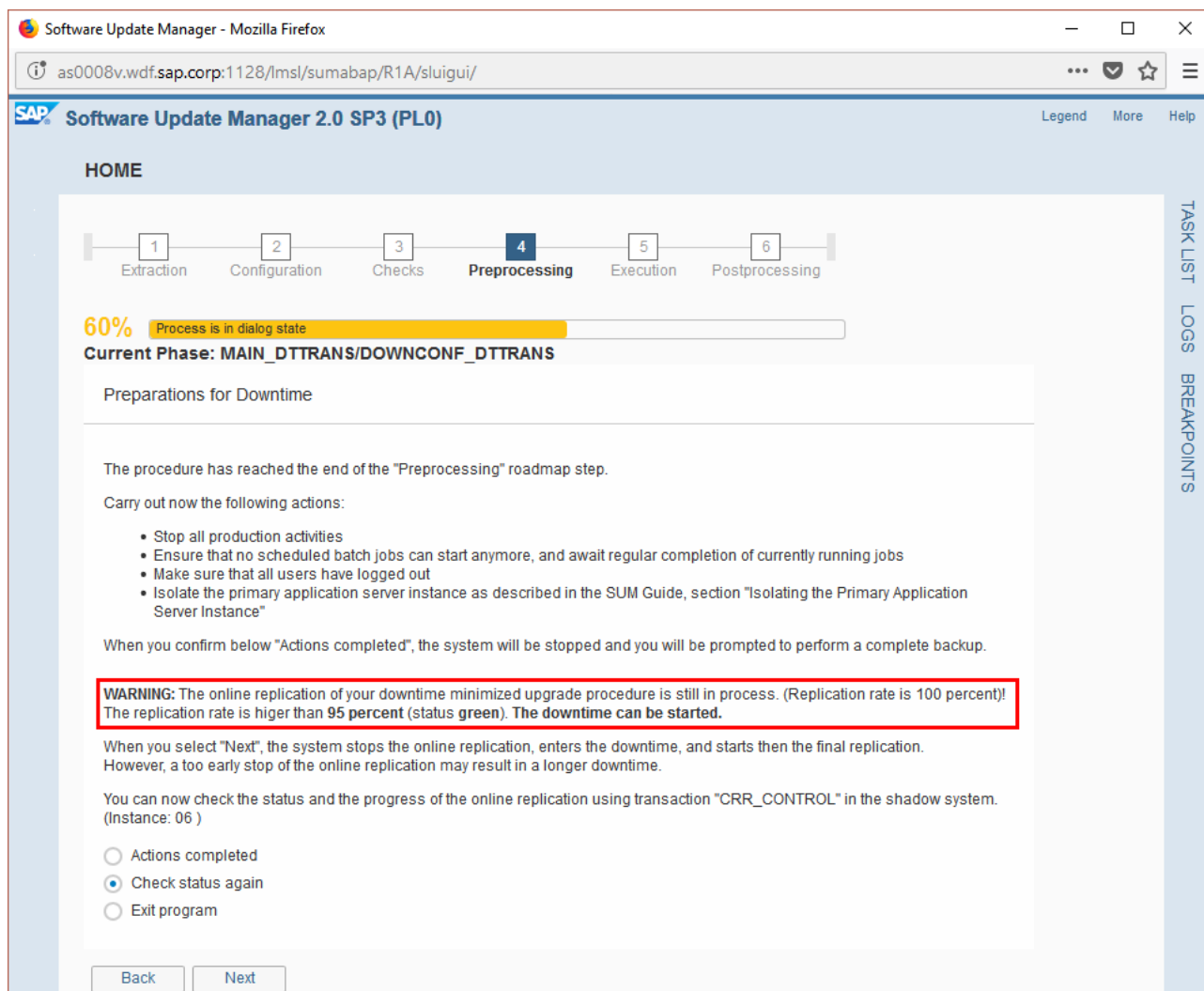


Figure 29: Check for at least 95% of replicated data before downtime

The replication itself runs in the shadow instance as mentioned in [figure 29](#). In contrast to a standard SUM procedure, the shadow instance is still active at this point of time. User DDIC can be logged on to the shadow instance to check the progress in detail with transaction `CRR_CONTROL`. Once the upgrade downtime can be started, first the standard instance will be stopped by the SUM, and afterwards the shadow instance to allow online data replication as long as possible. The remaining data is replicated explicitly during phase `RUN_RRC_REPLICATE_FINAL`. By using nZDM, the runtime of the technical downtime can be reduced to about 50% or more of the standard downtime. The real effect varies and depends on the system workload and number and type of involved tables. Technical details about this upgrade strategy can be found in the SAP blog article [Near-Zero Downtime Maintenance for SAP Business Suite Systems](#) with recommendations and customer examples.

There is also an nZDM technique that can be used for Java application server stacks, but this is currently not supported for platform IBM i. Therefore, also dual-stack systems, such as SAP Solution Manager or SAP NetWeaver Process Integration (SAP NW PI) systems cannot be upgraded on IBM i using the nZDM technology.

### 6.3 Zero Downtime Option

The next evolution of SAP upgrade tools tries to shorten the technical downtime of an SAP upgrade procedure even more than the near Zero Downtime Maintenance. The goal is to have no technical downtime and a minimized business downtime in order to keep the SAP users active while the underlying SAP system gets upgraded to a higher release. For this approach, a so-called bridge system will be installed in parallel to the affected SAP system with copies of all needed data in order to let the users continue their work. Currently, the Zero Downtime Option (ZDO) is not enabled for the IBM i platform.

### 6.4 Upgrade with Database Migration Option

The SAP upgrade tool SUM is able to run an upgrade combined with a database migration from any DBMS to SAP HANA Database. An IBM i server can be used to act as application server against an SAP HANA Database backend. In SUM phase `MIG2NDDB_INI`, the tool asks whether a migration should be done or not. The configuration options to manage the switch of the DBMS during the upgrade procedure are complex and a significant amount of work has to be done on the side of the new database server. This scenario would be worth a separate document and is, thus, not covered here.



## 7 Conclusion

As SAP evolved its large product portfolio over the years, all the needed tools to maintain the whole software product life cycle have been modernized as well to make your SAP system ready for current and future challenges. Software maintenance is an ongoing task for business-critical applications to keep them up-to-date from an application point of view and to provide the latest technical basis to run with. SAP provides information about the delivery of the latest [Software Logistics Tools](#) and available [maintenance targets](#).

The platform IBM i supports you running SAP applications as a reliable and powerful platform at lowest total cost of ownership. The recent [IBM i Technology Updates](#) are listed on the web at IBM DeveloperWorks. The latest POWER9 processor technology is available for IBM i to ensure optimal performance. IBM keeps driving a [strong strategy and a broad roadmap](#) for the IBM i platform to keep you prepared for future business.

As part of the Software Logistics Tools, the SAP Software Update Manager can perform a wide range of upgrade activities and it is likely that you will need to use it on several occasions over time. Understanding how SUM works, the important steps and how behaviors are shaped on your input should prove useful to you on those occasions in the future.

## 8 Table of Figures

Figure 1: SUM 1.0 for a dual-stack SAP system at the beginning.....	8
Figure 2: Temporary session settings for user QSECOFR.....	10
Figure 3: Directory structure of SUM version 1.0.....	13
Figure 4: Directory structure of SUM version 2.0.....	13
Figure 5: Running SUM processes in WRKACTJOB, overview with parallel SUMs for three SAP systems.....	17
Figure 6: Browser address bar with active secured HTTP access.....	18
Figure 7: The "More" menu.....	19
Figure 8: Progress view for ABAP upgrades and other GUI elements for SUM 2.0.....	19
Figure 9: SUM for ABAP stacks with active log file viewer.....	20
Figure 10: SUM Analysis viewer for ABAP upgrades.....	22
Figure 11: Process Control Center in a separate browser window.....	23
Figure 12: Roadmap steps in general.....	23
Figure 13: Pre-execution roadmap if a Java stack is involved.....	24
Figure 14: SUM upgrade evaluation form with phase runtime XML.....	25
Figure 15: ABAP shadow database library with views to shadow tables.....	27
Figure 16: Output of DSPFD for shadow view.....	27
Figure 17: Real name of the shadow table.....	27
Figure 18: Modifications in the SAP database during the upgrade with a shadow database.....	28
Figure 19: Java shadow database library with different table types.....	29
Figure 20: Correlation between a shadow view (as DDM file) and the table in the original database.....	30
Figure 21: First dialog of phase INITSUBST.....	33
Figure 22: Dialog options for parallel jobs in phase INITSUBST.....	34
Figure 23: Output of WRKJRNA command.....	37
Figure 24: Deleted Rows Analysis in Transaction DB02.....	46
Figure 25: Enable Parallelism in the Transport Management System.....	48
Figure 26: Decide an upcoming Unicode conversion during SUM.....	50
Figure 27: Enable downtime minimization methods in SUM.....	51
Figure 28: Turn on near Zero Downtime Maintenance.....	52
Figure 29: Check for at least 95% of replicated data before downtime.....	53

## 9 References

### 9.1 IBM Websites

#### 9.1.1 IBM Support and Knowledge Center

IBM Knowledge Center IBM i

[https://www.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i](https://www.ibm.com/support/knowledgecenter/ssw_ibm_i)

IBM Support Software Lifecycle (search for term "IBM I")

<https://www.ibm.com/support/home/pages/lifecycle/>

IBM Knowledge Center POWER9

<https://www.ibm.com/support/knowledgecenter/POWER9>

II14536 - SAP SUPPORT REQUIRED PTF LIST FOR IBM i 7.1 (InfoAPAR)

<http://www.ibm.com/support/docview.wss?uid=nas2II14536>

II14741 - SAP SUPPORT REQUIRED PTF LIST FOR IBM I 7.2 (InfoAPAR)

<http://www.ibm.com/support/docview.wss?uid=nas2II14741>

II14816 - SAP SUPPORT REQUIRED PTF LIST FOR IBM I 7.3 (InfoAPAR)

<http://www.ibm.com/support/docview.wss?uid=nas2II14816>

The IBM Toolbox for Java JDBC driver

[https://www.ibm.com/support/knowledgecenter/ssw\\_i5\\_54/rzamy/refjdbcdrv.htm](https://www.ibm.com/support/knowledgecenter/ssw_i5_54/rzamy/refjdbcdrv.htm)

RGZPFM - Basics on Reorganize While Active

<http://www.ibm.com/support/docview.wss?uid=nas8N1011025>

The IBM Insight for SAP utility

<https://www.ibm.com/partnerworld/page/techline/FAQ00000867>

#### 9.1.2 IBM Techdocs

SAP on IBM i - SAP Software Update Manager: Behaviors and Insights (this document)

<https://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP102762>

SAP on IBM i Recommendations and Review

<https://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101697>

SAP on IBM i Solid State Disk (SSD) Usage Recommendations

<https://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP102711>

### 9.2 SAP Websites

#### 9.2.1 SAP Community and Help Portal

SAP Support Portal

<https://support.sap.com>

SAP Product Availability Matrix

<https://support.sap.com/pam>

SAP on IBM i Community

<https://www.sap.com/community/topic/ibm-i.html>

SAP on IBM i Reference Architecture

<https://www.sap.com/documents/2015/08/fc338289-5b7c-0010-82c7-eda71af511fa.html>

SAP Software Logistics Toolset Delivery Notes

<https://support.sap.com/en/tools/software-logistics-tools.html>

SAP Software Logistics Toolset Support Page

<https://help.sap.com/viewer/p/SLTOOLSET>

Data Management Guide for SAP Business Suite

[https://help.sap.com/http.svc/rc/2eb2fba8f8b1421c9a37a8d7233da545/7.0/en-US/Data\\_Management\\_Guide\\_Version\\_70E.PDF](https://help.sap.com/http.svc/rc/2eb2fba8f8b1421c9a37a8d7233da545/7.0/en-US/Data_Management_Guide_Version_70E.PDF)

TCP/IP Ports of all SAP Products

<https://help.sap.com/viewer/ports>

### 9.2.2 SAP Wiki and Blogs

System Upgrade and Update

<https://wiki.scn.sap.com/wiki/display/SL/System+Upgrade+And+Update>

SUM for ABAP – Troubleshooting Guide

<https://wiki.scn.sap.com/wiki/display/SL/SUM+for+ABAP+-+Troubleshooting+Guide>

SUM Phases

<https://wiki.scn.sap.com/wiki/display/SL/SUM+phases>

Different types of Java upgrade

<https://wiki.scn.sap.com/wiki/display/Basis/Different+types+of+Java+upgrade>

SUM Java - Difference between Deploy based update and system switch upgrade procedures

<https://wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=358057306>

FAQ – near Zero Downtime Maintenance (nZDM) ABAP

<https://wiki.scn.sap.com/wiki/display/SL/FAQ+-+near+Zero+Downtime+Maintenance+%28nZDM%29+ABAP>

Explanations to the background and nature of SAP shadow instances

<https://blogs.sap.com/2014/04/02/sum-introduction-to-shadow-system/>

Description of feature Near-Zero Downtime Maintenance in SUM

<https://blogs.sap.com/2012/10/12/near-zero-downtime-maintenance-for-sap-business-suite-systems/>

Documentation about “Switch expert mode on” in the dialog of phase INITSUBST

<https://blogs.sap.com/2013/08/15/exploring-the-switch-expert-mode-on-feature-during-upgrades-and-ehp-implementations/>

Unicode Conversion Overview Guide

<https://blogs.sap.com/2009/04/08/unicode-conversion-overview-guide/>

Zero Downtime Option of SUM (ZDO) is “available on request”

<https://blogs.sap.com/2016/02/25/zero-downtime-option-of-sum-zdo-is-available-on-request/>

### 9.3 Other Websites

Toolbox for Java/JTOpen

<https://sourceforge.net/projects/jt400>

## 10 Abbreviations

CDS	Core Data Services
DB	Database
DBMS	Database Management System
DDM	Distributed Data Management
DMO	Database Migration Option
IASP	independent Auxiliary Storage Pool
IFS	Integrated File System
IGS	SAP Internet Graphics Server
ILE	Integrated Language Environment
JDBC	Java Database Connectivity
JSPM	Java Support Package Manager
JVM	Java Virtual Machine
LPAR	Logical Partition
nZDM	near Zero Downtime Maintenance
OS	Operating System
PAM	Product Availability Matrix
PASE	Portable Application Solutions Environment
PTF	Program Temporary Fix
REST	Representational State Transfer
RFC	Remote Function Call
SID	SAP system ID (a three-character identifier for an SAP system)
SPAM	Support Package Manager
SQL	Structured Query Language
SSD	Solid State Disk
SUM	SAP Software Update Manager
SWPM	SAP Software Provisioning Manager
TCP	Transmission Control Protocol
TMS	Transport Management System
UI	User Interface
URL	Uniform Resource Locator
ZDO	Zero Downtime Option

## 11 Trademarks

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world.

The following terms are registered trademarks of International Business Machines Corporation in the United States and/or other countries: AIX, AS/400, DB2, IBM, Micro Channel, MQSeries, Netfinity, NUMA-Q, OS/390, OS/400, Parallel Sysplex, PartnerLink, POWERparallel, RS/6000, S/390, Scalable POWERparallel Systems, Sequent, SP2, System/390, ThinkPad, WebSphere.

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries: DB2 Universal Database, DEEP BLUE, e-business (logo), GigaProcessor, HACMP/6000, Intelligent Miner, iSeries, Network Station, NUMACenter, POWER2 Architecture, PowerPC 604, pSeries, Sequent (logo), SmoothStart, SP, xSeries, zSeries. A full list of U.S. trademarks owned by IBM may be found at <http://www.ibm.com/legal/us/en/copytrade.shtml>.

NetView, Tivoli and TME are registered trademarks and TME Enterprise is a trademark of Tivoli Systems, Inc. in the United States and/or other countries. Oracle, MetaLink are registered trademarks of Oracle Corporation in the USA and/or other countries.

Microsoft, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

LINUX is a registered trademark of Linus Torvalds.

Intel and Pentium are registered trademarks and MMX, Pentium II Xeon and Pentium III Xeon are trademarks of Intel Corporation in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Oracle Corp. in the United States and/or other countries.

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.