In this sample, you’ll find a selection from Chapter 5, Upgrading the ABAP System. This chapter walks you through a full upgrade process from preparation to release, and serves as the base for most of the remaining upgrade chapters in the book.

Introduction
“Upgrading the ABAP System”

Contents

Index

The Authors

Mark Mergaerts, Bert Vanstechelman

Upgrading SAP
The Comprehensive Guide

573 Pages, 2015, $79.95/€79.95

www.sap-press.com/3631
Introduction

This book is about upgrading SAP systems, and SAP upgrades are something of a challenge.

SAP upgrades are challenging from a business perspective, because SAP production systems are critical for the functioning of a company, and therefore subject to very strict requirements as to availability, stability, and consistency. SAP software is also highly adaptable and customizable. Companies are not clones of one another, and neither are their SAP systems. Upgrading the SAP software must cause as little disruption as possible and must not interfere with changes or extensions that were designed to meet specific business needs. However, aiming for the status quo and refraining from upgrades altogether is not a viable option. Apart from the prosaic fact that every release at some time reaches the end of its support life (although SAP is more lenient in this respect than many other software vendors and maintains its product releases for many years), periodic upgrades are essential to keep abreast of crucial technological advances; the tremendous rise of mobile platforms and applications is but one example of this. Moreover, the rich functionality of a new release may boost a company’s fortunes and open previously unattainable opportunities. Like every other opportunity, this comes at a price: human resources must be engaged for planning, performing, and testing the upgrade, employee time is spent attending training sessions, and quite possibly new and more powerful hardware has to be purchased and installed.

Upgrades are also a technical challenge. In a distant past (“distant” in the IT sense of the word; let’s say up until the early 2000s), the reputation of SAP upgrades was fearsome, and for a SAP Basis consultant the first SAP upgrade was almost a rite of passage: do it right and you became part of the initiated. Consultants themselves did little to dispel this reputation and would be only too willing to entertain their public with tales of hard-won victories over the Upgrade from Hell.

In more recent years, the old magic has worn off a little, which does not mean that upgrades are now routine workaday tasks. Two influences are at work, tugging the level of complexity in opposite directions. On the one hand, the upgrade
tools themselves have gained much in power, reliability, and stability. Whereas in the old days there would be a lot of head scratching over arcane error messages, inevitably accompanied by long nights, sweet drinks, and greasy food; nowadays an uneventful, even dull upgrade is no longer an impossibility. On the other hand, the complexity of the SAP systems themselves has increased immensely. The monolithic SAP R/3 from olden days has been replaced with a bewildering array of specialized solutions; internally, SAP systems are now made up of multiple components, each with its own version track; and these SAP systems almost never operate in isolation but form parts of complex and constantly interacting landscapes in which changing one cog affects the entire machinery. Ask the old hands who wrote this book whether upgrades have become simpler, and their answer will be no. The difficulties are different and the pitfalls have moved, but they haven’t gone away.

This means that now, as in the past, upgrading SAP demands advanced knowledge in several fields: knowledge of the software components that make up the SAP environment, knowledge of the common technical platform (SAP NetWeaver) on which much of the SAP product line is built, and, last but not least, knowledge of the upgrade methods and tools themselves. Upgrades, therefore, belong in confident hands. By sharing with you their knowledge and experience, the authors—battle-scarred upgrade veterans—hope to bolster your confidence when you attack your next upgrade project.

This book is our third about SAP upgrades. Our earliest book on the subject was the mySAP ERP Upgrade Project Guide (SAP PRESS Essentials, 2005), which dealt with SAP ERP 2004 (SAP ECC 5.0), at the time the most recent SAP ERP release available and thus the most up-to-date successor to SAP R/3. This was followed by the much longer and ambitious SAP NetWeaver Application Server Upgrade Guide (SAP PRESS, 2007), which was based on SAP NetWeaver 7.0 and covered not only SAP ERP (then SAP ECC 6.0) but also other solutions, such as SAP Business Warehouse and SAP Supply Change Management; it also looked at the entirely new field of Java-based SAP systems. Saying that things have changed since 2007 would be a bit of an understatement; the tools are different, the products are different and more numerous, and in fact the entire concept of updating SAP software is also different.

This last point needs some explanation. One term that you will search for in vain in the 2007 book is “enhancement package.” By introducing enhancement packages for SAP NetWeaver and for the products that make up the SAP Business Suite (SAP ERP, SAP SCM, SAP CRM, and SAP SRM), SAP has moved from the massive “change everything” upgrades of the past to a more incremental approach in which new or enhanced functionality is introduced while keeping other components of the system stable. This more gradual approach of moving the software forward has important repercussions on the upgrade methods and tools, of course. As we will explain later in the book, SAP has unified the various change paths (release upgrades, enhancement package upgrades, and patches) into the concept of “updates.” Regardless of which component or components you change and how, the change is called an update and uses a uniform set of tools.

What Is in This Book?

The chapters in the book follow a line from the largely nontechnical to the purely technical. We start by placing SAP upgrades in a business perspective, looking at the incentives and caveats that surround upgrade projects. The subsequent chapters describe in detail the requirements, methods, and tools for upgrading SAP systems. You will learn how to prepare the SAP landscape for the technical upgrade, which upgrade tools exist, how they operate, and how to install them, get them running, and use them to control and monitor the process. We have concentrated the discussion of architecture and operation of the tools in a single chapter called A Guided Tour of the Upgrade Tools (Chapter 4). The concepts and procedures described in Chapter 4 are common to all upgrades, whatever the product, and thus form the basis for the specific upgrade procedures described in later chapters.

Following this guided tour are two chapters that deal with the two principal types of SAP systems: these chapters are Upgrading SAP: The ABAP System (Chapter 5) and Upgrading SAP: The Java System (Chapter 6). Both are independent of any specific product and provide you with the essential knowledge that you need to upgrade an ABAP-based or Java-based system. The information in these chapters is essential reading before you move on to the chapters that deal with specific components.

Chapter 7 covers what is probably the most critical action during the ABAP upgrade: the Modification Adjustment for SAP dictionary objects with Transaction SPDD. This is the point at which your understanding of the SAP system is put
to the test and at which human error can make the difference between a successful outcome and a dismal restore and try again. Understandably, this is also the step that invariably gives newbie upgraders butterflies in the stomach. Hopefully, working through this chapter will at least settle the butterflies, because SPDD is really not as bad as it seems.

With the foundations of upgrading laid, we move on to the part of the book that deals with upgrades of specific SAP components.

One can look at SAP products from two distinct angles: there is what we could call a functional view, which looks at the functionality the product provides regardless of the business purpose, and there is also a process view, which emphasizes the business processes that the product supports and facilitates. In the SAP world, we encounter the functional view in the context of SAP NetWeaver, where it is known as a “usage type.” Examples of usage types are SAP Business Warehouse (SAP BW; for data warehousing) or SAP Enterprise Portal (for company end user portals). The process view appears in the context of the SAP Business Suite, an ever-widening product portfolio of which the best-known members are SAP Enterprise Resource Planning (SAP ERP), SAP Supply Chain Management (SAP SCM), SAP Customer Relationship Management (SAP CRM), and SAP Supplier Relationship Management (SAP SRM).

The component-specific chapters cover the main SAP NetWeaver usage types: SAP BW, SAP Process Orchestration (SAP PO, formerly known as SAP PI and SAP XI), and SAP Enterprise Portal (SAP EP). The component-specific chapters also address the SAP Business Suite solutions SAP SCM, SAP CRM, and SAP SCM. For the SAP Business Suite, you might wonder why we seem to ignore the elephant in the room—namely, SAP ERP. The reason is that, technically speaking, an SAP ERP upgrade is largely or entirely covered by the standard upgrade tasks. The fact that the example upgrade used in Chapter 5 is for an SAP ERP system is perfectly rational. In a way, you could see an SAP ERP upgrade as the “default” upgrade, and an upgrade of other SAP Business Suite applications as “SAP ERP with extras.”

The component-based part of the book ends with a short chapter on upgrades of SAP Solution Manager. This is neither an SAP NetWeaver usage type nor a member of the SAP Business Suite (although in purely technical terms, it is an SAP CRM system).

The book concludes with a list of documentation references and some detailed technical data gathered in appendices.

---

**Terminology**

While writing the previous section, we felt ill at ease when using terms such as “product,” “application,” “solution,” and “component.” We are quite sure that we got them wrong somewhere, and an SAP marketing person reading our text would probably shake his or her head at our confused rambling. That is just one example of the fact that when it comes to rigorous use of terminology, SAP is a hard nut to crack. The product structure is complex; two terms can be synonyms or not quite synonyms; and the naming and numbering of versions seem to possess a logic that is all SAP’s own (and nobody else’s). We can only say in our defense that we do our best and try to navigate the quicksand of “SAP speak” with the greatest possible care.

A few terms used throughout the book merit closer attention:

1. The use of *update* versus *upgrade* is discussed at the beginning of Chapter 4. Suffice it to say here that for the version change scenarios covered in this book (changing to a higher product release and changing to a higher enhancement package within the same release) we consistently use the term “upgrade.” This is not entirely in line with the SAP usage of these terms, but it corresponds fully with the normal usage by SAP technical consultants and administrators.

2. SAP ERP is built around a core known as SAP ECC (Enterprise Central Component). Although “SAP ERP” undoubtedly expresses the business purpose of the software better and is also the name under which the product is described in SAP documentation, “SAP ECC” is more often used in technical contexts, especially if a version number is attached (e.g., one will encounter “SAP ECC 6.0” far more often than “SAP ERP 6.0.”). Because the target audience for this book is technical, we follow this same convention and thus use “SAP ECC” when referring to the technical component (which means most of the time) and “SAP ERP” only for the generic product.
5 Upgrading the ABAP System

Since the first release of SAP R/3 in 1992 to the latest and greatest enhancement package of SAP ECC, ABAP systems have been with us for more than two decades. In that time, the technology for release upgrades has become immeasurably more sophisticated, user-friendly, reliable, and powerful. But the ABAP systems themselves have also undergone a spectacular evolution. First, they have become much more complex (containing, for instance, many different software components, in contrast to the monolithic structure of R/3). Second, they have also become much larger. (We remember a time when a 200 GB system was considered big, whereas now multi-terabyte production systems are more the rule than the exception.) Finally, the ABAP systems, whatever their type (SAP ECC, SAP CRM, SAP SRM, or other) are now almost never standalone units, but are instead part of an integrated network of collaborating systems that consist of other first-line business components, data warehouses, portals, and the countless other applications that populate the modern corporate software ecosystem.

However, despite how much these upgrade tools may have improved, upgrading an ABAP system is still no simple matter. The earlier chapters on planning and preparation have already covered this aspect, so we’ll confine ourselves here to reminding you of a few crucial principles:

- **An ABAP upgrade requires expertise**
  
  There is always a first time and everybody must learn, but novices should build up this expertise by assisting in upgrades under expert supervision. Before touching their first upgrade, even under supervision, these novices should have
Upgrading the ABAP System

a good background in ABAP technology (with knowledge of the transport system and the SAP Data Dictionary essential).

- An ABAP upgrade requires time
  There is extensive preparation work required, both one-time project-wide work (such as identifying and installing the upgrade media) and repetitive, system-specific work. The technical upgrade itself runs for several days, and postupgrade activities, including testing, may also take up significant time.

- An ABAP upgrade requires resources
  With advancing technology, software becomes ever more efficient and code ever better optimized, but somehow this gain is almost always outweighed by the extra functionality that comes with a new release. Thus, every new version ends up using more computer capacity than the previous one. Also, the upgrade process makes demands of its own: it needs dozens of gigabytes of disk space (as we saw in Chapter 4), and while active it may use a good deal of processor power and I/O bandwidth. Starting an upgrade on server infrastructure that already has trouble coping with the load in the old release is a recipe for trouble.

- An ABAP upgrade requires collaboration
  Although steering the technical upgrade is your task and yours alone, there are other players in the game whose input or active assistance you will need. These include developers to do the adjustments to workbench objects (Transaction SPAU) or perhaps help you with the dictionary adjustment (Transaction SPDD); transport administrators to manage and import the queue of transport requests created by developers for the new version; functional specialists and key users to answer application-specific questions and perform testing; system administrators to provide needed computer resources and plan and execute backups; and management to ensure that the whole machinery runs smoothly and people remain focused. This does not mean (fortunately!) that technical upgrades are more about people management and social interaction than about “doing magic stuff” in the system; normally a project manager will be there to handle those aspects. Nevertheless, as the technical expert you must know who is who, who does what, and how you can call upon these people when you need their help.

The aim of this chapter is to take you through an end-to-end ABAP upgrade. We’ll start with the system preparations, then work our way through the entire technical upgrade process, and show you the postprocessing tasks that almost all ABAP upgrades require. However, before getting our hands dirty, there are a few important considerations to discuss and some concepts to introduce when planning an ABAP upgrade.

5.1 Planning the ABAP Upgrade
Before starting to prepare the upgrade technically, a few crucial decisions have to be made, including the following:

- What is the timeline of the project?
- Which upgrade strategy do we choose?
- Will we work in database archiving mode or not?

Figure 5.1 provides a top-level view of the upgrade depicted on a time axis. Times are relative to a point X, which denotes the start of downtime. The SAP system is production capable in the old release up to X and becomes productive in the new release at the start of the post-upgrade phase (shown as shaded rectangles at the bottom of the figure).
The pre-upgrade phase encompasses various preparatory activities, both long term and short term. It is difficult to give a precise starting point for this phase, hence the \( X - ?? \) denotation on the time axis. However, the work will start as soon as a release upgrade is being realistically considered. This does not mean that a firm decision to upgrade has been made by that point. The functional analysis, capacity-planning exercise, calculation of the financial impact, and other factors can still lead to the upgrade project being delayed or even cancelled. Assuming that the upgrade does take place, the activities in the pre-upgrade phase will become more and more specific as the start of the technical upgrade approaches.

The technical upgrade process is the actual transition of the SAP system from the old to the new release. This transition is carried out by means of SAP utilities, which run under control of the SAP-supplied upgrade control program, the Software Update Manager (SUM), which was introduced in Chapter 4.

When the SUM is finished, the system is operational in the new release, but not yet ready for normal business use. Custom modifications have not yet been reintegrated, code has not been regenerated, authorizations have not yet been adjusted, and so on. Various postprocessing actions are required to bring the system to a state at which it can be released for production use. While these actions take place, the system is inaccessible to all but a few key users, so this postprocessing phase must be considered as downtime (although technically the system is up and running). This is the reason that we include the postprocessing in the technical upgrade process.

The release of the system to its end users marks the beginning of the post-upgrade phase. Both the contents and duration of this phase depend on the type of system being upgraded. For development systems, the Modification Adjustment (bringing custom-made changes to SAP programs, screens, and so on back in line with the new release) will often be the most important task, possibly taking several weeks. In test systems, comprehensive testing of business processes on current or quasicurrent production data will dominate the post-upgrade activity. Like the Modification Adjustment in the development system, these tests may take weeks and involve significant human resources. In production systems, post-upgrade work should be limited mostly to end user support, monitoring and tuning, and minor problem resolution. Ideally, the duration of the post-upgrade phase in the production system should be near zero, although realistically even a well-prepared and well-performed upgrade will require a few days of “baby sitting” after the system goes live again.

In the next section, we will discuss the necessary timing involved when implementing the ABAP upgrade.

### 5.2 ABAP Upgrade Timeline

One of the most crucial questions in a technical upgrade is how it will affect the availability of the SAP system. In most places, taking down a production system is not an easy matter, and many of the advances in upgrade technology have been aimed at reducing the downtime and the resulting business impact to an absolute minimum. However, there is currently no such thing as a zero-downtime upgrade, and therefore timing considerations still play a very important role in the upgrade process. Given the potential difficulty these timing issues may cause, it’s important to address them as early and accurately as possible.

#### 5.2.1 Downtime before the Upgrade

Apart from the inevitable downtime during the technical upgrade itself, there may also be a need for downtime before the upgrade. This occurs specifically when the new SAP release requires an upgrade of the database software, the operating system, or both. Good planning and preparation will of course reveal at a very early stage the need for such pre-upgrade downtime, but accommodating it into a production schedule still can be difficult and can even lead to the upgrade itself being postponed.

An especially delicate situation arises when there are issues of compatibility between the current SAP version and the new version of the database or operating system that is to be installed. Suppose, for example, that a company has decided to upgrade from SAP version 1 to SAP version 2. This new SAP version requires the underlying database software to be upgraded from its own version 1 to a higher version 2. To reduce the downtime and technical complexity of the SAP upgrade and eliminate the risk of problems caused by changing several software components at the same time, the database upgrade should be done before the SAP upgrade. This means that during a transition period the system will have SAP version 1 running in combination with database version 2, as shown in Figure 5.2.
But what if SAP version 1 is not supported with DBMS version 2? Usually, SAP mitigates this problem by providing special support for this combination, but only as part of the upgrade trajectory—that is, the combination of SAP v1 and DBMS v2 is only supported in the run-up to the SAP upgrade. Although this solves the problem that a system must never be allowed to run an unsupported software release, companies might still feel that this combination of versions is less reliable than one that is supported unconditionally and therefore insist on keeping the transition period as short as possible. This may have a serious impact on the timing of the upgrade, because it is then necessary to have two downtime windows close to one another, something that might be possible only at a specific time of the year.

The worst-case situation is the one in which the new SAP release is truly incompatible with the underlying layers and the database software, the operating system, or even the hardware have to be changed simultaneously at the moment SAP is upgraded. Such a “big bang” approach increases the downtime, complexity, and risk associated with the upgrade. If a big bang cannot be avoided, then it must be planned and tested extensively and with the greatest care.

### 5.2.2 System Availability during the ABAP Upgrade

With respect to the availability of the system, we can divide the technical upgrade of the ABAP system into four parts, which we discuss in the following subsections:

- **Uptime**
- **Uptime, but without development or transports**
- **Downtime**
- **“Functional” downtime**

**Uptimes**

From a timing perspective, most of the upgrade happens in uptime: the system is available to end users and can be used productively. During this period, the objects of the new version are imported into the database. The upgrade process also carries out an extensive series of pre-upgrade checks and preparations.

One limitation, which is of fairly little relevance in production systems but much more so in development systems, is that some time into the upgrade the ABAP Workbench must be locked. From this point onward, it is no longer possible to change any development object (such as a program or table) in the system; also, all transport activity, both imports and exports, must cease. In exceptional and urgent cases, it’s possible to temporarily unlock the workbench, but whichever change is then made will not be included in the upgrade and so, if it is still applicable in the target release, will have to be reapplied after the upgrade.

**Downtimes**

The most critical parts of the upgrade, including the actual switch from the old to the new release (which is prepared during system uptime, in parallel with normal system use), take place in downtime. During this period, no activity other than the upgrade itself is possible. During the downtime phase, the upgrade locks the system to prevent users from connecting and it also autonomously shuts down and restarts the central instance as needed.

When the upgrade emerges from its downtime activity, it notifies the upgrade administrator via the SUM GUI that the system will now be returned to productive operation. The system is automatically restarted for the last time and then unlocked. However, “productive operation” should not be taken too literally at this time. First of all, the upgrade process itself is not quite finished, and the remaining activities may still take up a few hours. Although the work the upgrade has left to do is not critical, it is a very bad idea to let users work in the system as usual, because none of the postprocessing actions—some of which are critical—have been carried out yet.
During the downtime part of the upgrade, your role as upgrade administrator will mainly amount to sitting back and relaxing (or sitting back and dozing off if it’s the middle of the night), unless of course errors occur or some unforeseen intervention is suddenly needed. Once the upgrade finishes, however, this time of leisure comes to an end and some hard work awaits you. Between the end of the technical upgrade and the moment when the system is good for service and returned to the end users lies a series of postprocessing activities. Some of these activities are purely technical, which means you will be doing them yourself. Others require the intervention of the developers, transport administrators, or functional specialists. All this happens in what we call functional downtime; from a purely technical perspective the system is up and running, but in reality access to it is restricted to members of the upgrade team. This functional downtime also includes the time needed for post-upgrade testing.

5.2.3 System Resources and Upgrade Scenarios

Closely intertwined with the question of timing is the question of system resources. As we noted earlier, the technical upgrade is quite a resource-hungry process, and there is little point in running much of the upgrade during uptime if that slows down business transactions and batch jobs to a snail’s pace.

Early on in the upgrade, you’ll have to make a decision that essentially amounts to making a trade-off between downtime and resource usage: you will be able to save on the one by allowing more of the other. At the start of the configuration step (described in detail in Section 5.9), the upgrade process lets you choose between different upgrade scenarios. The basic choice you make here is between running the upgrade in a resource-minimized way or in a downtime-minimized way. If you go for the resource-minimized option, then the upgrade will use system resources as sparingly as possible—for example, by not running the main productive instance and the shadow instance (explained ahead) simultaneously. The price to pay for the lower capacity usage, however, is longer downtime. The downtime-minimized approach achieves the opposite: the downtime for the technical upgrade is reduced, even greatly reduced, but with a greater demand for computer resources.

The resource-minimized strategy provides the following features:

- The production and shadow system only operate independently of each other—in other words, only one of them is running at any given time. As a result, no additional resources are needed to support an extra instance.

- Production operation stops before the import of the new repository into the shadow tables or, at the latest, before the shadow instance is started for the first time.

- The Incremental Table Conversion (Transaction ICNV) is not used. All tables are converted during downtime.

The following are the features of the downtime-minimized strategy:

- Parallel operation of production system and shadow system.

- Import of the new repository into the shadow tables during production operation.

- Modification Adjustment of the ABAP dictionary objects (Transaction SPDD) takes place during production operation.

- Activation and distribution (both lengthy upgrade phases) are performed during production operation.

- Production operation stops at the latest possible point, in the DOWNCONF_TRANS phase (at the end of the Preprocessing roadmap step).

Which of the two strategies is the best? The answer to this kind of question is usually a more or less elaborate version of “it depends.” Not so, however, in the case of upgrade strategies—at least in our view. Here, we can without hesitation offer clear-cut advice: you should always use the downtime-minimized strategy.

Why so emphatic an opinion? Except in the unlikely case that a very small production system runs on a hugely powerful server and/or production downtime is of no concern, downtime minimized will normally be the only acceptable method for your production upgrade. One of the main aims of the upgrades of the test and development systems (where availability requirements are lower and resource minimized might be theoretically possible) is to prepare a correct and efficient procedure for the production upgrade. That really means that whatever you do in production you must already have done before. Of course, you can never eliminate the risk that a problem will occur for the first time in production, but you should not amplify that risk by choosing a different strategy for the production upgrade and thereby venturing into uncharted territory.

The argument of lower resource use and thus better performance with the resource-minimized strategy does not hold water either. In our experience, the shadow instance does not create excessive load on the system. A server with a reasonable amount of spare capacity should be able to handle the additional activity without
trouble. If the normal productive operation already loads the server to such an extent that it cannot bear the extra load of the shadow instance, then how can you expect it to cope with the new release and its increased resource requirements?

Finally, the downtime-minimized strategy truly delivers on its promise: it drastically reduces downtime. In most upgrades we have carried out with the currently available method of downtime reduction (the system switch method, described ahead), the downtime part of the technical upgrade mostly ran between five and eight hours. Longer runtimes are possible—for instance, due to large offline table conversions or long-running data conversion (XPRF) programs—but these will become apparent during the test upgrades. The shortened downtime makes it possible to plan the production upgrade during a normal two-day weekend. In a typical scenario, the upgrade would enter downtime late on Friday evening. The SUM would then be ready by Saturday morning (even allowing time for a backup), at which time postprocessing would begin. This should be ready by late afternoon, leaving Saturday night free for another backup, an update of the database statistics, and other tasks. Key users then have all Sunday to do their testing. If they give the green light, the end users will find the system ready for business on Monday morning.

5.2.4 Near-Zero Downtime Maintenance (nZDM)

If the features of the downtime-minimized strategy are still not enough and you want an even shorter upgrade downtime, then you can use Near-Zero Downtime Maintenance or nZDM. This technique has been available since SUM 1.0 SP 7 and can be used for all upgrade and update scenarios (release upgrades, enhancement package updates, and support package stack updates), although its benefit is greatest for the heavy release upgrades and EHP updates. With nZDM, SAP aims to bring the total downtime (upgrade and postprocessing combined) below four hours.

nZDM uses a technique known as Change Recording and Replication (CRR). The function of CRR is to record changes made by business transactions in the productive system and to replicate these to the shadow system. This mechanism is not needed for every table, but only for tables that are affected by the upgrade or update. Most of the recorded changes are replicated to the shadow tables in uptime; only the last 10% or so is replicated in downtime.

You can find more details about nZDM in the SAP upgrade guides and also in SAP Notes 1678564 and 1678565.

5.2.5 Backups

Backups are like bicycle helmets: many people dislike them and would rather go without them, until the time when something bad happens and they owe their lives (for the helmets) or their careers (for the backups) to having played by the rules. What makes backups particularly unpopular during upgrades is that they take precious time. This is made worse by the fact that these backups should really be made offline, either in the technical sense (with no activity at all in SAP or with the SAP system even shut down) or in a functional sense (with only a minimum of noncritical activity in SAP going on). When time becomes tight and deadlines loom, the temptation to skip a backup becomes greater and greater, especially because it is highly probable that you will get away with it.

A proper upgrade plan should, at a minimum, include backups at three points, as discussed in the following subsections.

Backup before Entering Downtime

A first backup is required at the end of the uptime and before the technical upgrade enters its downtime phase. This backup ensures that all business data is saved (remember that the system was in productive use up to this point). Also, the downtime part of the upgrade normally runs with database logging disabled; in some database systems, a backup is a mandatory precondition to be able to switch off database logging.

At this point, you must also make a backup of the SUM directory. This is because the files in the SUM directory are closely in sync with the upgrade content in the database. If something goes badly wrong in the downtime part, then having cross-consistent backups of the database and SUM directory enables you to resume the upgrade at the beginning of downtime.

Before entering downtime, the SUM will remind you to make both backups and to disable database logging, so there is no excuse for “forgetting” to make these backups now.

Backup after Technical Upgrade

This backup should be made after the end of the technical upgrade and before carrying out any dangerous postprocessing. By “dangerous,” we mean any activity that, in case of serious errors, might make the SAP system inconsistent and possibly
unchange. Individual post-upgrade actions do not get an official danger rating, so
this is basically a matter of good judgment and common sense. In our own upgrade
practice, the most critical point is the import of the upgrade transport queue (which
contains the modification transports created in Transaction SPAU as well as other
transports related to the upgrade). We always insist on a backup of the system
before this queue is imported, because of the risk of mistakes and resulting inconsis-
tencies. If such inconsistencies do occur, then identifying and fixing them can
take more time than simply restoring the backup and taking proactive steps (like
throwing out the “bad” transports) to prevent the problem in the first place.

Another dangerous activity is releasing the background jobs. At the start of down-
time, all background jobs—except the one that is necessary to manage transports
into the system—are suspended. This stops production jobs from inadvertently
running during critical parts of the upgrade. During post-upgrade activities, this
suspension should remain in place, because you don’t want to set the production
batch schedule loose in a system that isn’t yet ready and tested. Again, releasing
the jobs without a backup of the system would be irresponsible. In practice, this
is not really a big issue, because the release of the background jobs usually hap-
pens at the very end of the post-upgrade steps (and in many cases is done by the
system administrators and not by the technical upgrade team).

Testing is also potentially dangerous because test scenarios frequently include at
least some transactions that modify business data. A well-prepared test plan will
make sure that this does not have adverse effects on the consistency of the data,
but there is as always the possibility for things to go wrong. Because testing always
happens after the import of the transport queue, the pre-import backup already
offers protection. However, if enough time is available, an additional backup
between the end of postprocessing and the beginning of testing will do no harm.

Note
In some cases, the functional tests make irreversible changes to the data, in which case
a backup is made just before testing starts and then restored after testing is complete.
This is an extremely safe but obviously time-consuming way to work. The people who
are responsible for the test plan must be able to say whether this approach is advisable
or necessary.

Other activities—for example, generating the ABAP loads with Transaction
SGEN—carry very little if any risk, and there is normally no problem running
those before the post-upgrade backup or even in parallel with an (online) backup.

Backup before Productive Start

Finally, a backup is needed after all upgrade activities, including testing, have
been completed and the system is ready to be returned to its end users. This
backup is the very last stage in the technical upgrade process and ensures that a
clean, fully upgraded copy of the system is backed up and can, if necessary, be
restored without the need to redo any of the post-upgrade work and testing.

After this final backup, the normal day-to-day backup policy for the system can
resume.

5.2.6 Downtime after Go-Live

Once the system moves to a productive state, there is no longer any need to stop it
or restrict access to it. The main task for the upgrade and functional teams is now to
monitor the system and to act on the problems they discover. We know of upgrades
in which everything ran perfectly from the word go, but in most cases at least a few
issues will crop up, with short dumps, performance complaints, and interface prob-
lems making up the top three issues in our experience. The majority of these issues
can be dealt with online; a major exception, however, is when the system suffers
from less-than-optimal parameter settings. This can happen for database parame-
ters, but more often the problem arises with SAP profile parameters, and more spe-
cifically with the parameters for buffer sizing and memory management.

As we said before, every new release needs more resources than the last, and this
is certainly true for the memory used by the SAP application servers. Therefore,
Transaction ST02 (which shows buffer and memory usage) is certainly one of the
most important monitoring tools in the days after the upgrade. If you see that
memory performance is poor (with buffers or memory areas filling, lots of object
swaps, and in general a Transaction ST02 display showing more and more red),
then it will be necessary to increase the affected buffers and memory areas.
Unfortunately, most profile parameters controlling these sizes are static and thus
need a stop and restart of the SAP instances to take effect.

During upgrade postprocessing, we always try tuning the SAP parameters to the
best of our ability, based on experience, pre-upgrade memory statistics, and data
from previous upgrades in the same landscape. Still, predictions of the behavior
of the productive system are never more than approximate, so there is a genuine
possibility that the buffers will need to be tuned after the go-live. For this reason,
we always try to negotiate an optional downtime window in the first days after
the upgrade. This could be during the following weekend, for example; unless the problems are so severe that the system becomes unworkable, it’s normally possible to survive for a week with the suboptimal parameters. This downtime is short; because you can prepare the new parameter settings online in Transaction RZ10, you only need time to stop and start the instances of the system, which should not take more than, say, 15 minutes for a single-instance system and 30 minutes for a system with multiple instances.

In our experience, obtaining this extra downtime window for tuning is almost never a problem, but you must of course make sure this is properly agreed on in advance.

### 5.2.7 Time Schedule during the Technical Upgrade

To assist you in setting up a workable schedule for the production upgrade, we’ll provide a practical example of how to time the technical upgrade process. The schedule listed in Table 5.1 assumes that the downtime-minimized strategy is used and that a database backup takes four hours. The schedule is comfortable, in the sense that there is a large safety margin for the completion of the uptime part of the SUM (36 hours, from Thursday 6 a.m. to Friday 6 p.m.), and there is even slack time during the upgrade weekend.

<table>
<thead>
<tr>
<th>Time</th>
<th>Uptime/Downtime</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saturday (week 1)</td>
<td>DOWN</td>
<td>➤ Upgrade DBMS (if necessary).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Set database and SAP profile parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(e.g., <strong>DIR_PUT</strong>) to values required for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>upgrade (if necessary).</td>
</tr>
<tr>
<td>Monday 9 a.m.</td>
<td>UP</td>
<td>➤ Perform the Initialization SUM step.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ In the Configuration step, indicate that</td>
</tr>
<tr>
<td></td>
<td></td>
<td>you want to slow down the import by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>specifying a number of import processes smaller than 1.</td>
</tr>
<tr>
<td>Tuesday 9 a.m.</td>
<td>UP</td>
<td>➤ Preparatory steps of SUM (up to and including the Checks step) are finished. Start the Preprocessing step.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Repository is imported (slowed down).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Development and transports are locked.</td>
</tr>
</tbody>
</table>

Table 5.1 Example Schedule for Technical Upgrade (Cont.)

<table>
<thead>
<tr>
<th>Time</th>
<th>Uptime/Downtime</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wednesday 9 a.m.</td>
<td>UP</td>
<td>➤ Import ends.</td>
</tr>
<tr>
<td>Wednesday 12 noon</td>
<td>UP</td>
<td>➤ Dictionary modification adjustment (Transaction SPDD).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Start activation.</td>
</tr>
<tr>
<td>Wednesday 6 a.m.</td>
<td>UP</td>
<td>➤ First activation run ends (with errors).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Process activation errors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Restart activation.</td>
</tr>
<tr>
<td>Wednesday 9 p.m.</td>
<td>UP</td>
<td>➤ Activation ends.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ SUM starts remaining pre-downtime phases.</td>
</tr>
<tr>
<td>Thursday 6 a.m.</td>
<td>UP</td>
<td>➤ SUM reaches the downtime switch point.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Let the process wait.</td>
</tr>
<tr>
<td>Friday 6 p.m.</td>
<td>DOWN</td>
<td>➤ Stop user activity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Suspend background jobs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Isolate system.</td>
</tr>
<tr>
<td>Friday 6:30 p.m.</td>
<td>DOWN</td>
<td>➤ Create backup.</td>
</tr>
<tr>
<td>Friday 10:30 p.m.</td>
<td>DOWN</td>
<td>➤ Start SUM downtime part (Execution step).</td>
</tr>
<tr>
<td>Saturday 5:30 p.m.</td>
<td>DOWN</td>
<td>➤ SUM downtime ends; final steps (Postprocessing and Finalization) are executed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Switch database log mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Create backup.</td>
</tr>
<tr>
<td>Saturday 9:30 a.m.</td>
<td>DOWN</td>
<td>➤ Postprocessing (technical).</td>
</tr>
<tr>
<td>Saturday 2:30 p.m.</td>
<td>DOWN</td>
<td>➤ Import transport queue.</td>
</tr>
<tr>
<td>Saturday 5:30 p.m.</td>
<td>DOWN</td>
<td>➤ Development and functional postprocessing.</td>
</tr>
<tr>
<td>Saturday 10 p.m.</td>
<td>DOWN</td>
<td>➤ Create backup.</td>
</tr>
<tr>
<td>Sunday 2 a.m.</td>
<td>DOWN</td>
<td>➤ Update database statistics.</td>
</tr>
<tr>
<td>Sunday 8 a.m.</td>
<td>DOWN</td>
<td>➤ Key users start testing.</td>
</tr>
<tr>
<td>Sunday 6 p.m.</td>
<td>DOWN</td>
<td>➤ End testing: give final go/no-go.</td>
</tr>
<tr>
<td>Sunday 7 p.m.</td>
<td>DOWN</td>
<td>➤ Create backup (restore in the case of no-go).</td>
</tr>
<tr>
<td>Monday 7 a.m.</td>
<td>UP</td>
<td>➤ Unlock users.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Reschedule background jobs and open interfaces to the external systems.</td>
</tr>
</tbody>
</table>

Table 5.1 Example Schedule for Technical Upgrade (Cont.)
5.3 The Shadow Repository and Shadow Instance

Let’s set originality aside for a moment and quote literally from the ABAP upgrade guides. Our only contribution is set the key points in bold:

*The Software Update Manager updates your system using a system cloning and switch procedure.* This procedure installs a copy of the system, the shadow system, in parallel with the original system. The shadow system is used to update the affected software components and to install the additional components, while the original system is still in production operation.

What SAP is describing here is what makes it possible to run the greater part of the upgrade in normal uptime with the system productive.

Over time, SAP has made various improvements and tweaks to the system switch, and this is still the method in use today. This means that when you upgrade an ABAP system, at some time you will see that, in parallel with the SAP central or primary instance, there is a second instance on the same server. You will even log on to this shadow instance and work in there, namely to make the dictionary adjustments (Transaction SPDD) and to fix possible activation errors. In this shadow instance, you are already working with the new version while the main system is still running the old one. You now know how SAP works this bit of magic: the shadow instance works with the shadow repository while the main system continues to work with the old repository.

The shadow instance needs no extra management or clean-up effort. The upgrade process creates the shadow instance, starts and stops it automatically when necessary, and dismantles it when it is no longer needed. No trace of it remains after the upgrade is finished. Although the SUM fully manages the shadow instance, there can be rare situations where human intervention is needed (for example, after an unplanned system failure). Command-line methods are therefore available to do things such as starting or stopping the shadow instance manually should this ever be necessary.

5.4 The ABAP Upgrade Directory

The ABAP upgrade creates its own directory structure below the SUM directory. The path is simply `<DIR_SUM>/abap` (note that abap is in lowercase).

Several subdirectories reside below abap. Figure 5.3 shows the contents for a recent SAP ECC 6 EHP 7 upgrade.

![Directories for ABAP Upgrade](image)

We won’t bore you with a description of each subdirectory; you can find that information in the SAP ABAP upgrade guides if you are interested. However, a few things do merit attention.

First of all, if you look at the list of subdirectories, you will see names such as buffer, cofiles, data, log, or sapnames. Do these look familiar? Yes indeed, those are subdirectories you also find in the standard transport directory. Of course, that is not a coincidence; much of the ABAP upgrade process is based on importing transport requests, and in that context the subdirectories of the SUM serve the same purpose as their namesakes in the transport system. Stretching things a bit, you might say that an ABAP upgrade is a gigantic import operation, although this would not be entirely accurate—for example, the shadow repository is imported with database-level tools and not by using the programs of the transport system.

Monitoring the upgrade and finding information if things go wrong is of course one of your principal tasks as the technical upgrader. You can do this in the SUM GUI via the menu option ABAP • Logs, but to be honest we rarely ever use this and prefer to follow the logs directly at the server level, where we can use operating system commands to help the analysis (for example, text-finding commands, such as `grep` in UNIX/Linux or `FINDSTR` in Windows). Unless you have no O/S level access to the server (making your upgrade work quite uncomfortable!) we recommend that you do the same. In that case, the following knowledge is essential:
The upgrade logs are gathered in the subdirectory `log` (no surprise there).

Upgrade phases that use the transport system create their logs in the subdirectory `tmp`. At the end of the phase, the completed log is moved to the log subdirectory. This means that you find the log(s) of the currently running upgrade phase in most cases in `tmp`.

Phases that do not use the transport system normally create their logs directly in the `log` subdirectory; these phases do not use `tmp`.

The main logfile of the technical upgrade process is a file called `SAPup.log`, which you find in the `log` subdirectory. This file is so important that we return to it in the next section.

A trace of all the interaction that takes place in the SUM GUI, including messages displayed by the upgrade and input provided by the user, is kept in `SAPupConsole.log`, also in the `log` subdirectory. This file thus provides an audit trail of all decisions made during the upgrade and all feedback received from the upgrade process. As an illustration, we have provided a brief extract from this file (see Listing 5.1).

```
>> 2014/04/21 13:00:12 START OF PHASE PREP_PRECHECK/SPAMCHK_INI
=========== SPAM Version Check ===========
SPAM version 52 in your source system is sufficient for this procedure. Nevertheless, we recommend that you import the latest SPAM update. The program has not found a SPAM Update for release 701 in the EPS inbox.
SPAM update message: No SPAM/SAINT update found
01) - Search for newer SPAM version in \mysaptst\sapmnt\trans\EPS\in
   02) * Skip SPAM update
: Skip SPAM update
```

Listing 5.1 Trace Interactions in the SUM GUI

In this example, the upgrade reported the current version of the SPAM tool in the source system and asked whether a higher version should be searched and applied or the current version kept. In reply, the upgrader chose to keep the current version by selecting the option `Skip SPAM update`.

Finally, a few more things that are useful to know:

- The ABAP upgrade directory contains both a `bin` and an `exe` subdirectory. They have different purposes: `bin` contains the executables, scripts, and other files for the upgrade process itself; `exe` contains the new SAP kernel.

The `htdocs` subdirectory contains documentation files—for example, the HTML report generated at the end of the upgrade and showing data such as phase execution times is created here. These are files you may want to transfer to your workstation and keep for later reference.

Also at the end of the upgrade, a backup (in zipped form) of the upgrade logs is made in a dedicated part of the transport directory, at `/usr/sap/trans/UPGRADE`. This means that the logs are preserved even after the SUM directory itself is discarded (as it is bound to be at some point after the upgrade). However, because in many installations the transport directory has a habit of filling up and therefore needs regular cleaning, overzealous system administrators might decide to throw away these backups. Therefore, we recommend that you keep a copy of these ZIP files yourself. Logs from previous upgrades, even old ones, have sometimes proven very valuable to us, so don’t throw them away.

### 5.5 SAPup

The program responsible for the ABAP upgrade (and started by the SUM server) is called `SAPup`. `SAPup` maintains a high-level logfile containing the flow of the upgrade; this file, `SAPup.log`, is therefore the main log of the entire upgrade. It does not contain all the details; you find those in the individual phase logs (which together amount to several gigabytes and many millions of text lines by the time the upgrade reaches its end). However, `SAPup.log` is essential, because it contains information that gives you a complete view of the upgrade process from beginning to end.

For every upgrade phase, it contains the following information:

- Start time of the phase
- Begin and end time of interaction with the upgrade administrator
- End time of the phase
- End result of the phase (succeeded, failed, or skipped)
- Extra information (for some phases)

Let’s look at a few extracts from the `SAPup.log` from a real upgrade. First, the most ordinary sort (see Listing 5.2).
Upgrading the ABAP System

CURRENTPHASE PREP_GENCHECKS/NTACT_CHK
...started at 20140420183912
Ø Using phase log file 'NTACT_CHK.LOG'.
..finished at 20140420184235 with status SUCCEEDED.

Listing 5.2 Ordinary SAPup.log Upgrade Phrase

This phase ran successfully and did not interact with the upgrade GUI. The timestamps show that its runtime was 3 minutes and 23 seconds. Because this is not a phase that uses the transport system (and therefore does not name its logs according to the transport system naming rules), SAPup.log also gives the name of the phase log file (see Listing 5.3).

CURRENTPHASE PREP_PRE_CHECK/SPAMCHK_INI
...started at 20140409130012
# Using phase log file 'SPAMCHK.LOG'.
...begin dialog at 20140409130018
...end dialog at 20140409130225
..finished at 20140409130225 with status SUCCEEDED.

Listing 5.3 SAPup.log Phase Log File Name

Again, we are looking at a phase that ran problem free, but here some interaction went on with the user manning the SUM GUI. You can see from the consecutive timestamps that the phase took just over two minutes, but all but six seconds of this was spent waiting for an answer from the upgrader (see Listing 5.4).

CURRENTPHASE MAIN_NEWBAS/XPRAS_AIMMRG
...started at 20140411185540
..finished at 20140411185654 with status FAILED.
Ø Error message set: 'Detected 12 errors summarized in 'XPRASUPG.ELG'
Calling 'K:\usr\sap\TST\DVEBMGS01\exe/tp' failed with return code 8,
check Z:\SUM\abap\log\SAPup.ECO for details'
...begin dialog at 20140421185702
...end dialog at 20140421194508
..answered at 20140421194508.
-> decided to try again.

CURRENTPHASE MAIN_NEWBAS/XPRAS_AIMMRG
...started at 20140421194508
..finished at 20140421194641 with status SUCCEEDED.

Listing 5.4 Failed SAPup.log Upgrade Phase

This one didn’t go so well. The phase first ended with status FAILED and produced an error screen in the SUM GUI (begin dialog). The problem clearly needed some time to investigate and fix (or the upgrade administrator happened to be away or not looking), because the user’s answer came 48 minutes later. At that time, the upgrader requested to repeat the phase. This worked well, and the new run of the phase ended without errors after a little under two minutes.

The previous log extract also shows that this phase was transport based (it used tp). The transport process reported a return code of 8, which (as you ought to know, with your knowledge of the ABAP transport system) means that at least one object in the transport request caused an error.

The last example is seen in Listing 5.5.

Ø Phase JOB_RSUPDTEC skipped due to condition(s) 'DBTYPE=ORACLE DBTYPE=DB6'
CURRENTPHASE [PREP_INIT/JOB_RSUPDTEC]
...started at 20140409130405
..finished at 20140409130405 with status SKIPPED.
Ø Phase GEN_TAIASAP skipped due to condition(s) 'DBTYPE=DB6 DBTYPE=ORACLE'
CURRENTPHASE [PREP_INIT/GEN_TAIASAP]
...started at 20140409130405
..finished at 20140409130405 with status SKIPPED.

Listing 5.5 Non-Executed Consecutive Phases

Here, two consecutive phases were not executed at all. SAPup.log gives the condition that led to the phases being skipped. In this case, the upgrade phases only apply to SAP systems running on Oracle or on DB6 (the SAP acronym for DB2 on UNIX/Windos). Because this was an upgrade on a system with SQL Server, the phases were duly skipped.

SAPup Command-Line Options

Let’s take closer look at some of the command-line options that you can use with SAPup. These options enable you, for instance, to change upgrade parameters or to control the shadow instance if for some reason your express intervention is needed. You will not always need these features, but they can be quite useful in problem situations.

You can execute SAPup with a command-line option in two ways:

1. In the SUM GUI, choose ABAP • START WITH OPTIONS.
2. Use the command line in a terminal session (UNIX, Linux) or in a CMD or PowerShell session (Windows).
The SAPup program is located in the directory `<DIR_SUM>/abap/bin`. To display the list of all available options use `-h`—for example:

```
cd /usr/sap/PRD/SUM/abap/bin
./SAPup -h
```

Some options are fairly dangerous and should only be used in emergencies and under the guidance of SAP Support. The ones described in the following paragraphs are generally harmless and are those which, speaking from experience, you are most likely to use.

**Unlock and Lock the Shadow Instance**

The upgrade process itself handles locking and unlocking of the shadow instance. When locked, it is only possible to log on as SAP* or DDIC; any attempt to log on with a different user is rejected with the message "Upgrade still running; logon not possible." The upgrade unlocks the shadow instance at the beginning of the activation phase to let you run Transaction SPDD and also in case of activation errors to enable you to correct these errors.

In exceptional circumstances, you may have to lock or, more likely, unlock the shadow instance manually. This is done via the following command-line options:

```
SAPup [rootdir=<DIR_SUM>/abap] lockshd
SAPup [rootdir=<DIR_SUM>/abap] unlockshd
```

The optional `rootdir` parameter specifies the path to the ABAP upgrade directory. We recommend that you always include it.

**Start and Stop the Shadow Instance**

Even more rarely, the shadow instance may have to be started or stopped explicitly, which you can do via the following command-line options:

```
SAPup [rootdir=<DIR_SUM>/abap] startshd
SAPup [rootdir=<DIR_SUM>/abap] stopshd
```

**Check the Shadow Instance State**

You can check whether the actual state of the shadow instance matches the expected state for the currently active upgrade phase via the following command-line option:

```
SAPup [rootdir=<DIR_SUM>/abap] checksysstatus
```

**Set Upgrade Parameters**

As we saw earlier, you provide the parameters for the upgrade through input windows displayed in the SUM GUI. There may be situations in which you want to change these parameters later. A common example is that during the upgrade, someone changes the password of the DDIC user in client 000 (which you had to supply to the SUM in the Extraction roadmap step), something which causes the upgrade to fail with an invalid password error the next time it tries to log on to SAP. Using an SAPup command option, you can supply the new password to the SUM so that the upgrade can continue.

The commands to change parameters should always be run from inside the GUI and not in a terminal session, because they expect a GUI connection (see Table 5.2).

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>set stdpar</td>
<td>Standard parameters for upgrade</td>
</tr>
<tr>
<td>set shdpar</td>
<td>Parameters for the shadow instance</td>
</tr>
<tr>
<td>set confpar</td>
<td>Upgrade configuration parameters (e.g., number of parallel processes)</td>
</tr>
<tr>
<td>set ddicpwd</td>
<td>DDIC password in main instance</td>
</tr>
<tr>
<td>set shddicpwd</td>
<td>DDIC password in shadow instance</td>
</tr>
</tbody>
</table>

**Table 5.2  SAPup Options for Upgrade Parameters**

**Force Upgrade to Stop at Next Phase**

You can use a SAPup command-line option to instruct the upgrade to stop at the beginning of the next phase. This might be useful if, say, the host computer is misbehaving—for example, performance is poor because of excessive paging, and you decide to reboot the host.

To do this, use the following command-line option:

```
SAPup [rootdir=<DIR_SUM>/abap] stop
```

Now that we have looked at the program used for our upgrade, let’s get started!
5.6 Starting the ABAP Upgrade

You now have enough of technical and practical background to get going, so now is the time to embark on a real ABAP upgrade.

Note

Most screenshots in this chapter come from a real production upgrade from SAP ECC 6.0 EHP 4 to EHP 7. The system, which had multiple instances, ran on Windows servers. The underlying database system was Microsoft SQL Server. Although we used the occasion to collect screenshots for this book, we could not, of course, produce every possible screen, because this was a production system. Situations such as creating a breakpoint in the upgrade were fairly harmless, but provoking unforced errors just to show the effect was obviously out of the question. Therefore, some screenshots come from “dummy” upgrades in which we were free to steer off course.

The screens displayed by the SUM and the various interactions with the user depend on the upgrade context and may therefore vary. We cannot guarantee that the screens you will see and the input you will have to provide will be exactly the same as what you see in the screenshots. However, we have tried to include a maximum of screenshots by also using material from other upgrades.

5.6.1 Prerequisites

Before we get going, let’s briefly go over a few important prerequisites.

Stack XML file

Don’t even dream of starting the upgrade before you have a stack XML file that the upgrade accepts as consistent. There are few nastier surprises than having a stack XML file rejected by the SUM when the project clock is already ticking. We have known of situations in which, even with the help of SAP support, it took weeks to finally coerce the Maintenance Optimizer into producing a correct file, so this is one area in which you do not want to take even the slightest risk.

You must place the stack XML file at the top level of the download directory. This is the directory where you have placed the upgrade DVDs and also the files that you selected and downloaded in the Maintenance Optimizer. This is the same as the "media directory" we talked about in Chapter 3, but here we refer to it as "download directory" because that is the term used in the SUM.

System Access

To do a decent job, you need decent access. In the SAP system, you must be able to log on to client 000 and to the default (productive) client. For the upgrade, we normally create a dedicated user (or request one to be created if, for example, the system is under Central User Administration) to be used exclusively for the upgrade. We do this even in systems in which we have a personal-user ID, because a user specifically made for the upgrade makes all upgrade-related tasks (such as transport requests) easily identifiable in the future.

Ideally, this upgrade user should have the SAP_ALL privilege so that it does not run continuously into authorization errors. Our own practice is to create this user as a copy of DDIC or SAP*. In the address information, the upgrade user should be clearly linked to the person responsible for the technical upgrade, and the user’s access should be limited in time with the valid-to date set not too long after the planned end of the upgrade project.

In any case, the user ID to be used for the upgrade and the roles or profiles assigned to it must be discussed with the SAP system administrator. You must not fall foul of the company’s security policy or SAP’s user licensing rules.

You should also be able to access the upgrade server at an operating system level. This can be a problem in companies in which the infrastructure is managed by an outsourcing partner; in such outfits server access is often restricted to the outsourcing company’s staff. If you find yourself in this situation, first try to obtain a temporary access to the upgrade host. If that is denied, then the outsourcing partner will have to make the necessary resources available to execute operating system tasks (which can be as simple as starting the SUM) on your behalf. Keep a close eye on their responsiveness, though; some outsourcers answer quickly and efficiently, but there are others who insist on all the bureaucratic niceties. Waiting half a day just for someone to type "STARTUP" is no fun; don’t sit on your frustration if this is happening! Instead, raise a timely alert.

During upgrades, almost all server-level tasks are performed as the <sid>adm user. On UNIX and Linux, root access is needed for some smaller operations, especially in relation to the new SAP kernel. It is quite likely that you even if you have O/S access to the server you will not be granted the right to log in as root. In that case, a UNIX system administrator must be available to carry out these tasks.
On Windows hosts, the `<sid>adm` account always has local administrator privileges, and there are no upgrade tasks that need domain controller rights; therefore, you should not run into any restrictions here.

**Passwords**

As we saw earlier, the SUM will prompt you for several passwords, which it stores in encrypted form. One password that you will always have to supply is that of the user DDIC in client 000. The SUM will ask for additional passwords depending on the operating system and database type: with upgrades on Windows, for example, you will be asked for the passwords of the two SAP users `<sid>adm` and `SapService<SID>`; for Oracle databases, you must supply the password of the SYSTEM user. Take note of which passwords the SUM needs (but please don’t put the passwords in your documentation; we see this surprisingly often), and make sure you are given these passwords in time for each upgrade. If possible, avoid changing any of these passwords during the upgrade, but if a password does get changed, then there is a command-line option in SAPup that lets you communicate the new password to the SUM.

**tp and R3trans**

It is highly recommended to install the latest versions of the `tp` and `R3trans` programs of the source kernel release before you start the upgrade. The upgrade process itself requires a minimum version. If the installed versions are too low, then the SUM will complain at a very early stage (see Figure 5.4).

**Update SPAM/SAINT**

As for `tp` and `R3trans`, it’s also advisable that you install the latest available version of Transactions SPAM and SAINT into the SAP system you are about to upgrade. If the source release is SAP NetWeaver 7.0 or higher, then you need the Maintenance Optimizer for this, because the package must be approved in your download basket. For lower start releases, you can download the SPAM/SAINT package directly from the SAP Service Marketplace. The name of the package is `KDxxxyy.SAR`, where “xxx” denotes the SAP Basis version and “yy” the patch level.

To install the patch, log on to client 000 as a user different from SAP* or DDIC (the upgrade user you may have created earlier in 000 is a good choice), and call Transaction SPAM. If you have extracted the package yourself into the EPS/in directory on the SAP transport host, then choose `Support Package /L54263 Load packages /L54263 From application server`. Alternatively, if you have the KD* archive on your PC, choose `Support Package /L54263 Load packages /L54263 From frontend`. Next, choose `Support Package /L54263 Import SPAM/SAINT update` and reply to the pop-ups SPAM displays next.

The update typically takes 5 to 10 minutes and can safely be done while the system is in normal use. Sometimes a short dump occurs towards the end. This is normally harmless; simply restart SPAM and confirm the update.
5.6.2 Software Update Manager Roadmap Steps

We can now start the SUM server and SUM GUI. You will find the instructions for this in Chapter 4, Section 4.2.6.

The ABAP upgrade is divided into major stages, known as roadmap steps. In the first upgrade, there are eight such steps:

1. Initialization
2. Extraction
3. Configuration
4. Checks
5. Preprocessing
6. Execution
7. Postprocessing
8. Finalization

Because these roadmap steps are the milestones of the upgrade, we’ll come back to them regularly. At this point, it is sufficient to say that the first five steps (up to and including Preprocessing) happen in uptime; Step 6, Execution, runs in downtime, and Steps 7 and 8 run in what we have called functional downtime, with the SAP system operational and unlocked but not yet ready for business use. The next sections follow through these steps, beginning with Initialization.

5.7 Initialization Roadmap Step

On the GUI login screen, choose the Administrator role, identify yourself, enter the password, and optionally give a telephone number or email address where you can be reached. Then, click OK (see Figure 5.5).

The Welcome window is now displayed. No input is requested here, but do a quick check that the information about the SAP system is correct, and then click Next (see Figure 5.6).
In the **Specify Credentials** screen, enter your credentials in the **User Name** and **Password** fields, and then click **Next** (see Figure 5.7). This window may differ depending on the operating system of the upgrade host.

On the next screen, shown in Figure 5.8, you enter the path to the Stack XML file created for this upgrade (see Chapter 3).

As the SUM screen explains, the stack XML file that you specify must be located in the download directory. The SUM does not prompt you separately for the path to that directory; it assumes it will find all downloads it needs in the directory in which the stack XML file is located.

On this screen, there is also a **Manually prepared directory** option. This is only used for updates of single Java components and does not apply here; you always need a valid stack XML file for an ABAP upgrade (or Java upgrade, for that matter).

Click **Browse**, and select the stack XML file that was generated for this system in the Maintenance Optimizer (see Figure 5.8).
Upgrading the ABAP System

No luck: the SUM did not like the stack XML file that you gave it. The error information on the screen is not of much use, but there is a reference to a log file that will hopefully tell you more. Note that at this point the logs are still in the SUM’s own SDT directory and not yet in the ABAP subdirectory.

Let’s look at the file DEFINE-TARGET-SOURCE_01.LOG (see Figure 5.10).

Things could have been much worse than this. The error simply states that the stack XML file referred to another host (actually, it had been copied from the stack XML file of an earlier upgrade).

If you play strictly by the rules, you should now create a new stack XML file for this specific system. In this case, we had established that the installed software components in every system of this SAP ECC landscape were absolutely identical, so we created one stack XML file and copied this for use in the other systems. In these copies, we then edited the system-specific elements (system name, central instance number, and server name). To be honest, this is not a practice SAP recommends; SAP frowns on manually editing stack XML files. This is mostly with good reason, but here the changes were fairly trivial and easy to identify, and we wanted to avoid another battle with the Maintenance Optimizer.

After correcting the stack XML file, we resubmit it to the SUM, and this time we are successful (see Figure 5.11).
Upgrading the ABAP System

Keyword; the screen shows in which note to find this. As we explained before, the purpose of asking for a keyword is to give the SUM at least some reason to believe that you have properly read the upgrade note before starting, so please consider this as a gentle reminder that the note really is essential reading.

Type the keyword (it is shown in plaintext, because it’s publically available), and click Next.

Everything we have done up to this point is part of the Initialization step (see Figure 5.12), which has now come to an end. To move on to the next step, click Next.

Figure 5.12 End of the Initialization Step

Note

By choosing Back instead of Next, you can undo the entire roadmap step and restart it from the beginning. This option will also be available at the end of other (though not all) roadmap steps.

5.8 Extraction Roadmap Step

This is the first roadmap step of the actual ABAP upgrade. The Initialization step was handled by the SUM directly, but now the SUM runs the SAPup process to work. From this point on, logs and other files are created in `<DIR_SUM>/abap`. The SAPup.log file is also created now.

This is also the first step to be subdivided into phases, most of which run unattended. Although the upgrade is active without the need to interact with the SUM, the GUI does show the list of phases (see Figure 5.13).

Figure 5.13 Phase Display while Upgrade is Active

A green square indicates a completed phase, and a gray diamond indicates a phase that is currently active or still has to begin. The name of the active phase appears at the bottom of the screen. This is also the phase name you will find in SAPup.log. The phase name consists of two parts: the module name and the individual phase name, separated by a forward slash. Within a roadmap step, there may be several modules.
**Upgrading the ABAP System**

SCAN\_DOWNLOADDIR is actually an important phase, because at this point the SUM examines the ABAP download directory to make sure everything it needs is there. This phase runs unattended and takes some time (10 to 20 minutes is a good guess, but circumstances may vary).

**Tip**

In your upgrade documentation, always note the length of time the upgrade runs unattended. This helps you plan for the next point at which the upgrade will need input. Having timing data from previous upgrades also gives you a base for comparing between systems; if, for example, the unattended runtime in a certain system becomes much longer than what you observed in earlier upgrades in that landscape, then this could mean something is going wrong (like a hanging or looping phase) or that the server is experiencing performance problems.

At the next interaction point, the SUM asks for more passwords, including that of DDIC in client 000. The prompt for the SAP Service user is specific for Windows. Click **Continue** (see Figure 5.14).

The SUM now checks the installed version of SPAM/SAINT. In this case, we installed the latest version for the source release before the upgrade, so no further change is necessary. Nevertheless, the SUM gives you a chance to update SPAM/SAINT if you haven’t done so before (see Figure 5.15).

The screen informs you that the currently installed version is sufficiently high and that no further SPAM/SAINT update package is available in the EPS Inbox. This is fine; select **Skip SPAM update**.

The production system being upgraded here has its database and SAP primary instance running on different servers. In such a case, the SUM will issue extra prompts about the database server. First, the SUM asks you for the operating system on the database host (see Figure 5.16).
Because this system uses SQL Server, the list of possible operating systems only includes Windows. For database systems also supported on other platforms, such as Oracle, Sybase, or DB2, the list will be longer.

Next, you must specify the password of the database administration account on the database server. Again, this dialog appears only if the database is running on a different host (see Figure 5.17).

If everything goes correctly, then no further interaction takes place, and the upgrade runs unattended up to the end of the Extraction roadmap step.

At the end of every roadmap step that performs actual upgrade work in the system, a screen is displayed with a summary of the check results (see Figure 5.18). The messages appear in truncated form on the left side of the window. Click on a message to see additional information in the right-hand pane.

If there are no errors, then closing this window brings the upgrade to the end of the roadmap step (see Figure 5.19).
In exceptional cases in which you need to repeat the entire EXTRACTION step, you can trigger a rerun by clicking on Back. Normally, however, you want to move forward rather than backward, so press Next.

5.9 Configuration Roadmap Step

As the name implies, this roadmap step sets up the configuration that will be used for the upgrade.

5.9.1 Upgrade Strategy

Right at the start of the CONFIGURATION step, you are asked to choose the upgrade strategy.

In Section 5.2.3, we described the two strategies (resource-minimized and downtime-minimized) and pointed out their importance. As you can see in Figure 5.20, you are given a choice between not two, but three options. These are called configurations on the SUM screen, whereas the upgrade guides refer to them as pre-configuration modes.

The SINGLE SYSTEM configuration corresponds to the resource-minimized strategy. Fewer resources are used, mainly by never running the productive instance and shadow instance at the same time, at the expense of increased system downtime.

The STANDARD and ADVANCED configurations are both for the downtime-minimized strategy. The difference is that the STANDARD mode tries to reach a balanced compromise between downtime and resource usage, whereas the ADVANCED mode aims strictly for downtime reduction.

There are two more input fields in this screen as well. Selecting the Yes checkbox for the KEEP ARCHIVING ON DURING THE WHOLE PROCEDURE option has the effect that transaction logging at the database level remains active during the downtime phases of the upgrade (in the uptime part transaction logging will always be active, because there is concurrent business activity in the system). The exact mechanism differs between databases—for example, in Oracle it is known as ARCHIVELOG mode—but the effect is always that, in case of a recovery from a backup, all changes made by the upgrade can be replayed, leading to a consistent and fully (or partially) upgraded system.

Upgrades produce a massive volume of transaction logs, and leaving archiving enabled throughout the downtime part is very rare. Its usefulness can also be questioned: if a crash occurs during the upgrade, then you will probably want to return the database to the state it was in at the beginning of downtime, and then repeat the upgrade from that point. If the crash happens after the upgrade is complete, it is far more efficient to recover from a backup made after the upgrade than to have the recovery process reapply all changes made by the upgrade.

By selecting the SWITCH EXPERT MODE on box, you will gain much more control over the configuration of the resources the upgrade can use. You will find more information on this in a moment.

What did we choose in this example?

First, we chose a downtime-minimized approach. This will hardly surprise you: we have never used, nor do we ever want to use, resource minimized, and in this specific upgrade downtime was definitely a concern. Therefore, we chose the ADVANCED configuration.

For the reasons explained previously, leaving database archiving (transaction logging) enabled throughout the entire upgrade was not an option, so we left that box blank.
Finally, we like to be in control of things as much as possible; also, we tend not to trust “intelligent” decisions made by software. Therefore, we did tick the Switch EXPERT MODE on box. These decisions are shown in Figure 5.21.

Figure 5.21 Upgrade Strategy: Advanced Scenario and Expert Mode

5.9.2 Configuring the Downtime-Minimized Strategy

The next screen is quite long (make sure you use the vertical scrollbar to see what is at the bottom), and we have cut it into several pieces for presentation here.

The first part of the screen asks about Near-Zero Downtime Maintenance Technology (nZDM). This topic was introduced in Section 5.2.4. If you decide to use nZDM, then you must indicate how many CRR (Change Recording and Replication) processes to run in the background (see Figure 5.22).

Figure 5.22 Configuration: nZDM

We did not use nZDM, so this part of the input screen remained unchanged.

The next part of the input screen lets you configure the number of processes of different types that can run in parallel. For each process type, you can set separate values for uptime and downtime (see Figure 5.23).

Figure 5.23 Configuration: Number of Processes

The groups have the following meanings:

- **BATCH PROCESSES (UPTIME)**
  The number of background jobs launched by tp. Be careful with the uptime setting for this value, because enough jobs must remain available to run normal production batch jobs. It is sometimes a good idea to configure and activate an operation mode that provides additional background work processes.

- **SQL PROCESSES (DOWNTIME)**
  This refers to processes, also launched by tp, that execute DDL statements in the database (for example, creating new tables or indexes).

- **R3TRANS PROCESSES (UPTIME)**
  As you know, R3trans is the workhorse of the transport system. It is the program responsible for importing the content of transport requests into the SAP system. In many cases, it is possible to speed up the import of large transports by distributing the work over multiple R3trans processes.

- **R3LOAD PROCESSES (DOWNTIME)**
  R3load is another data mover, this one responsible for importing the new ABAP repository into the shadow system. A special feature here is that you can specify a fractional number of processes for R3load. Doing so has the effect of...
artificially slowing down the repository import. You might want to do this in busy production systems, because the data import causes significant load and generates huge amounts of database logging. In the SUM guide, SAP gives an indication of the expected run time if you set the number of R3load processes to 1; for example, the guide at the time of writing (November 2014) quotes a figure of six hours. You can reduce the impact of the process by entering a value smaller than 1; this will then lengthen the duration of the import proportionally. For example, to slow down the repository import to 24 hours you would set R3LOAD PROCESSES (UPTIME) to 0.25.

PARALLEL PHASES
Some upgrade phases can be run in parallel, and this is the maximum number of simultaneous phases you want to allow.

The next part of the screen lets you specify how background jobs are to be distributed over the different instances of the system. As the text on the screen makes clear, this applies only to the uptime part. During downtime, only the central instance (CI)—or primary application server (PAS), as it is now more correctly called—is available, so every background job runs there.

Note also the remark that the SUM directory must be accessible on every server involved in the processing of upgrade batch jobs (see Figure 5.24).

![Figure 5.24: Configuration: Uptime Background Servers](image)

Here, we accept the default choice, which is to leave the scheduling of the background jobs to the system.

In the next section, you can indicate that you want to run the shadow instance (which by default runs on the server of the CI/PAS) on a different host (see Figure 5.25).

To be honest, we have never used this option, and we do not intend to. In our view, the only reason for using this feature would be fear of a lack of resources on the central server, and, as explained earlier, upgrades should not happen at all on servers that are potentially short on capacity.

The generation of the new ABAP loads is one of the heaviest postprocessing operations. The generation may take several hours, during which the system is so heavily loaded that not much else can be done in it. Rather than scheduling the generation yourself using Transaction SGEN after the end of the upgrade, you can ask the upgrade process itself to take care of this (see Figure 5.26).

Once more, we like to stay in control of things, and we are so used to running Transaction SGEN after upgrades that we choose to do it ourselves. However, this is purely a matter of preference, and there is no objection to letting the SUM handle the SGEN; if you choose that option, however, we would probably advise you to configure more than three processes for it.

The last part of the screen (the bit you’ll miss if you forget to scroll down) lets you choose whether to use Incremental Table Conversion (Transaction ICNV) if any
large tables must be converted or whether not to use ICNV and let conversions happen in downtime (see Figure 5.27).

Figure 5.27 Configuration: Incremental Table Conversion

Skipping ICNV would be contrary to the downtime-minimized approach, so we definitely keep the ICNV option here. Note that this does not mean that an ICNV will really be necessary; the system doesn’t decide that until later.

After supplying all of these configuration parameters, you can finally click Next. If you are working in a system with several dialog instances, then one extra window will appear (see Figure 5.28).

Figure 5.28 Update Dialog Instances

Here, you are asked whether the dialog instances should be updated with the new kernel and restarted at the end of downtime. If you don’t tick this option, then the dialog instances will be stopped at the beginning of downtime and the upgrade process will not touch them further, leaving you the task of updating their kernel and restarting them after the upgrade. This might make sense if, for example, the existing dialog instances will not be used again after the upgrade, perhaps because you plan to install new ones. Otherwise, it is better to leave this chore to the SUM, which is what we do here.

5.9.3 Package Inclusion

The upgrade now runs for a substantial amount of time (for the production upgrade shown here, it was close to 90 minutes). Most of this time is spent in the phase EHP_INCLUSION, in which the system calculates the enhancement packages to be processed in the upgrade. A later similar phase does the same for the add-on components. The next point of interaction comes when this last phase asks you to indicate what should be done with the add-ons found in the system, shown in the screenshot of our sample upgrade (Figure 5.29).

Figure 5.29 Add-On Selection

The ST-A/PI is normally installed in every SAP ABAP system, so you will see at least that component listed here. Which other add-ons, if any, appear here varies between systems. For each add-on, the upgrade determines a default decision, which you see in the STATUS column. If you agree with this decision, then there is nothing to do; simply confirm the screen without selecting any of the checkboxes. Selecting a box is only necessary if you disagree with the decision proposed by the SUM. This is not normally something you would want to do, unless the upgrade note for the add-on tells you to do so (again, do look at the note referenced on the screen). However, there are situations in which the upgrade process cannot come to a satisfactory decision—for instance, because it cannot find the package it needs to upgrade the add-on. In that case the STATUS column shows UNDECIDED, and the checkbox is already selected for you. Figure 5.30 shows an example from another upgrade.

Figure 5.30 Add-On with UNDECIDED Status
This was an upgrade from a system based on SAP NetWeaver 6.40 upgrading to SAP NetWeaver 7.3. The ST-A/PI is a fairly simple add-on, which is not upgraded in the traditional way; you simply let the old version be deleted and then install the new version after the upgrade. This meant of course that the SUM could not find an upgrade package for it in the download directory and returned an UNDECIDED verdict.

For an UNDECIDED add-on, the next screen will present a list of possible actions (see Figure 5.31).

The correct decision depends on the add-on; for the simple case of the ST-A/PI, this was to delete it (an operation known as passive deletion). In most cases, however, deleting an add-on or allowing it to stay on the same version is not possible without causing major consistency problems. This is why the other DELETE and KEEP decisions require either a CD or SAINT package provided by the supplier of the add-on or a special key code, the VENDOR KEY. In those cases in which you are allowed to use a key or in which a CD or SAINT package is available, the add-on note will clearly tell you so. If you run into an add-on for which the decision is unclear and you have neither an upgrade package nor a key, then your only option is to open an incident with SAP support and ask for a course of action.

One message you should not send to SAP is “I have this add-on I don’t know how to handle, so please give me the vendor key so that I can keep the add-on and get on with the upgrade.” SAP Support will not let you corrupt your system, so they will not just throw a key at you; instead, they will try to figure out the proper solution. This might of course take time, but fortunately this is an issue you will encounter in the very first test upgrade. Take into account that such trouble with add-ons, especially the more exotic ones, does happen from time to time, so it is good practice to make a list of components in the system well before the upgrade, check out the upgrade notes about them, and, if anything is unclear, open a support incident up front asking for guidelines.

Some add-ons require a keyword; like the upgrade keyword, this is simply a basic control ensuring that you have read the upgrade note. For example, the keyword request for an upgrade of the BI Content add-on is shown in Figure 5.32.

After you have handled the add-ons, the upgrade will also ask you to confirm the support packages that will be bound into the upgrade. The SUM shows a window with the complete list of components; for each component, it shows the calculated target SP. Figure 5.33 shows just the first part of the list that was displayed in our example upgrade.

If the STATUS column is empty, then the upgrade is happy with the selection, and no further action is needed. An example in which the upgrade is not so happy is shown in Figure 5.34.
For the ST-A/PI, the calculated target level was SP 02, but the upgrade could only find SP 01 in the download directory. This case was pretty straightforward; because SP 01 met the minimum requirement, it was OK to keep it as the target patch level.

Other situations can be more serious, though; these are usually caused by the fact that not all patches were correctly loaded into the download directory. In such a case, the SUM will give you the chance to place the support package in the directory. It will then rescan the directory, load the missing packages, and then display a hopefully error-free patch list.

### 5.9.5 Customer Transport

One final prompt the SUM will display talks about a Customer Change Request (see Figure 5.35).

This feature is very rarely used. It’s possible in exceptional cases to bind a customer transport into the upgrade: mind you, not just any transport, but one that meets conditions prescribed by SAP and serves a very specific purpose (an old method of downtime reduction back in the days of release 4.6 used this; that is the only example we can remember of this function being used).

### 5.9.6 Requests for SPDD and SPAU

The upgrade will again run unattended for some time (in the example upgrade, almost one hour) before it stops to ask about transport requests for Modification Adjustment.

Modification Adjustment, the process of reconciling custom changes made to SAP repository objects and the new versions of those objects that come in with the upgrade, has two major parts:

- Adjustment of dictionary objects (tables, structures, data elements, and domains) with Transaction SPDD
- Adjustment of other objects (programs, screens, etc.) with Transaction SPAU

Both SPDD and SPAU save all actions taken on affected objects in a transport request, and the SUM is able to import that change request automatically instead of you having to repeat the Modification Adjustment again in every upgraded system. This is not mandatory; for SPDD it is up to you whether to use a transport from a previous upgrade or to repeat the manual work. For SPAU, which is typically handled by more than one person, the adjustments may be spread over multiple transport requests, which can be imported in the normal manner after the upgrade.

If you decide that a transport is to be used for the Modification Adjustment, then you must explicitly register the number of that transport at the end of SPDD or SPAU. The IDs of the transports are stored in a file in the transport directory.

At this point in the Configuration roadmap step, the SUM gives you the possibility to specify the SPDD and/or SPAU transport that it will import later. There are two possibilities here. The first is that it cannot find a reference to an SPDD or SPAU request, either because this is the first upgrade in the landscape or because you did not bother to register the transports of the Modification Adjustment during the previous upgrades. The screen then looks like Figure 5.36.
The second possibility is that the upgrade process did find a reference to a modification adjustment transport, or even more than one, in which case it lets you choose between the ones that it has found (see Figure 5.37).

**Figure 5.37** Modification Adjustments: Requests Found

Here, the SUM lets you choose between two SPDD requests created during the upgrade of the DEV and TST systems, specify a transport request of your own (in case an SPDD transport exists but you forgot to register it), or none at all (meaning you will perform the SPDD manually, which is what we chose here). In the example, no transports had been registered for SPAU, so no similar list was displayed for the SPAU requests.

Why did we decide to do the SPDD again manually rather than let the SUM do the work for us? In our case, the answer is that we have been working with SAP for many years, and in the early days we felt uncomfortable with letting the upgrade simply replicate the SPDD of one system in another system. The dictionary adjustment with SPDD is a delicate operation, because mistakes can lead to data loss, and we simply were not prepared to relinquish control of the process. For instance, what if an object had been modified in the test system in a different way than in the development system? The reality nowadays is that the upgrade handles this perfectly well; if a modified object is found that is not present in the SPDD transport or that has been modified in a different way, then you will still have to adjust that object manually with SPDD. Therefore, there is really no objection any longer to using a Modification Adjustment transport of a previous system. But in our case, old habits die hard.

5.9.7 **Shadow Instance**

The last thing the Configuration step will ask for is the number of the shadow instance. Because the shadow instance will run in parallel with the primary instance, at least in the downtime-minimized scenario, it must have a number different from that of the primary instance and from that of any other SAP instance running on the same server (see Figure 5.38).

**Figure 5.38** Shadow Instance

Note that this screen may prompt for extra information depending on the underlying database. In some databases (for example, Oracle) the shadow repository is installed with a separate schema so as to keep it separate from the data used by the productive system. For Oracle, the SUM will ask you to set a password for the schema user, and the screen will look like Figure 5.39.

**Figure 5.39** Shadow Instance: Oracle Shadow Schema

For the shadow system, the ports for the dispatcher and gateway services must be present in the SERVICES file of the host. For example, if the number chosen for the shadow instance is 51, then the following service entries are required:
On Windows servers, this is not an issue: the upgrade runs with local administrator rights, so it is able to update the SERVICES file itself. On UNIX and Linux, however, changing the /etc/services file can only be done by a root user. If the ports do not already exist, then at the end of the CONFIGURATION roadmap step the upgrade will report an error and request that you create the ports (see Figure 5.40).

Figure 5.40 Shadow Instance: Missing Service Entries (UNIX/Linux)

Sites with strict security policies will probably not let you work as root (and outsourcers certainly will not), so make sure a system administrator is available to do this for you.

There is one last question about the shadow instance before the CONFIGURATION step reaches the end (see Figure 5.41).

Figure 5.41 Shadow Instance: Reuse Profiles

This is another feature you are unlikely ever to use. If, during a previous upgrade, you decided for some reason to alter the parameter profiles for the shadow instance and you want to take over those settings in the shadow instance of the present system, then you can place the profiles you want to reuse in the save subdirectory below the main ABAP upgrade directory. The SUM will then take over these profiles when it sets up the shadow system.

Changing shadow instance parameters is very rarely necessary, and, even then, you might prefer to note this in your documentation and apply the changes manually.

5.9.8 List of Upgrade Files
The very last thing the CONFIGURATION step shows you is the list of all files that the upgrade has found in the download directory and will use for upgrading the system. Figure 5.42 shows a small fragment.

Figure 5.42 List of Files to Be Used in Upgrade

For each file, you see the name, the type, the way it will be handled, the status, and the description. For some files, it is fine that the status is empty rather than OK; this is true, for example, for the stack XML files.

5.10 Checks Roadmap Step
The next major step in the process, CHECKS, ensures that the system is in a proper state to be upgraded. During the CHECKS step, the system is still in uptime. Technically, this is still a preparatory phase. Apart from the upgrade tools themselves, nothing related to the upgrade is yet imported or adapted in the system; that will happen only in the next roadmap step.

5.10.1 Saving Variants
This operation may or may not be performed, depending on the upgrade context. In our example upgrade (SAP ECC 6 EHP 4 to EHP 7), it was not, so the screenshot of Figure 5.43 comes from a different upgrade.

In every SAP system, developers and even end users will create numerous variants with selection values for SAP programs. Because SAP programs and their
selection screens often change between releases, the portability of these custom-made variants can be problematic. The upgrade tools contain a set of programs to save the custom variants for SAP programs before the upgrade and to restore them afterwards, checking whether any new fields have been added. Saving the variants happens at this point in the upgrade (see Figure 5.43).

The phase JOB_RASUVAR1 starts a background job, which saves the variants. Depending on the number of custom variants, the runtime may vary from just a few minutes to several hours. Towards the end of the upgrade, a companion phase, JOB_RASUVAR2, will restore the variants.

5.10.2 The ASU Toolbox

For the next stage of the SUM, you had better roll up your sleeves, because there might be a lot of work to be done. The upgrade now wants you to start the ASU Toolbox (see Figure 5.44).

ASU stands for application-specific upgrade. Depending on the applications that are installed and used in an SAP system, there are always application-dependent actions to be performed both before and after the upgrade. In the past, information about these steps would be in the upgrade guide, in the upgrade note, in notes pointed at by the upgrade note, in notes pointed at by notes pointed at by the upgrade note, in notes pointed at... well, you get the idea. The risk that some of those actions would be overlooked was substantial, and this could lead to problems later. The answer SAP found to this problem was to provide an “umbrella” transaction that would list all the necessary actions and enable the user to invoke these actions from a single interface: this is the ASU.

At this point, the upgrade is asking you to carry out the pre-upgrade ASU (a similar request for the post-upgrade ASU will be made later). Now, as the name implies, the ASU actions are application specific, which means that you, being a technical person, will not necessarily understand what they are about, let alone carry them out all by yourself. This implies that the ASU is a collective effort in which you have to involve functional specialists. Even if you do not do everything yourself, you will still have to give these specialists some training or documentation on how to use the ASU transaction, which is the reason for our initial advice to roll up your sleeves; in our experience the ASU is one of the most labor-intensive parts of the whole ABAP upgrade.

The first step, preparing the ASU for use and finding out how it works, is your task regardless, so here goes:

1. Log on to the SAP system in the default (production) client.
2. Start Transaction /ASU/UPGRADE.

You now see a list of all actions. In our example upgrade, this list was quite short (in fact, it was a rare case of the pre-upgrade ASU taking just a few minutes), so what we show in Figure 5.45 is the rather more impressive list from a BI upgrade (from version 3.5 to 7.3).

The question mark in the Status column means that you have not yet carried out this action, or at least have not yet assigned a status to it.

The type of action is denoted by the icon to the left of the description. In most cases, you can call the program or transaction by double-clicking; for other activities double-clicking brings up a text window in which the necessary actions are described. You can display further information for every activity by clicking the icon in the Description column and can view the corresponding SAP Note by clicking in the Note column.
After you have checked and possibly executed an activity, you must set its status by clicking on the question mark icon in the Status column. A window then pops up in which you choose the appropriate status to be assigned (see Figure 5.46).

The initial status of an activity is always STATUS NOT YET DEFINED (question mark). At the end of the process, no more question marks should be present; that is, you must have assigned a status to each activity. It is quite normal for the worklist to contain activities that are not relevant to this specific system and therefore can be skipped.

Our normal approach to the ASU is this:

- Call each action in the list. If it is technical, then carry it out as instructed (or mark it as PROCESSING SKIPPED if irrelevant) and note the procedure in the upgrade documentation so that in later upgrades it becomes routine.
- If the action is functional and you see that you do not have the knowledge to perform it correctly, then put it on a to-do list for the functional team.
- Give the complete to-do list to the functional team together with instructions on how to use Transaction ASU and/or a brief training session.
- Also ask the functional team to document the procedure; include their documentation in your own upgrade document. This is useful if the same person is not available to repeat the task during a later upgrade or, in the worst case, you have to perform the task yourself.
- As always, note the time each action takes. The runtime of ASU activities can be anything between near zero and several hours, so it is essential that you know how much time will be spent on it in coming upgrades. Timing information must be collected in a test upgrade of a production-sized system, not of the development system.

Sloppy handling of the ASU can lead to a substantial loss of time, so make sure that the procedure is properly described during the first test upgrade.

### 5.11 Preprocessing Roadmap Step

The request to process the pre-upgrade ASU is the last phase of the CHECKS roadmap step. As always, the SUM shows you a summary of errors (in which case you have to repeat the step) or other facts of interest. If no errors were found, then you move on to the next step, PREPROCESSING.

This is the part of the upgrade where the real things finally start happening. During the PREPROCESSING step, the new repository is imported, the shadow instance is configured and started, you are requested to perform Modification Adjustment on the dictionary with Transaction SPDD, the new dictionary is activated, enhancement packages and support packages are imported into the shadow
# Contents

Acknowledgments ............................................................................................ 19
Introduction ..................................................................................................... 21

## 1 Project Planning .................................................................................. 27

1.1 Why Upgrade? ....................................................................................... 27
  1.1.1 New Possibilities, Features, and Functionality ........................ 28
  1.1.2 Outdated SAP Version .............................................................. 28
  1.1.3 Release Support and Maintenance Costs ............................... 28
  1.1.4 Installing Support Packages versus Upgrading ...................... 29
  1.1.5 To Unicode, or Not to Unicode: That Is the Question .......... 29
  1.1.6 SAP HANA ........................................................................... 29
  1.1.7 Upgrading Is a Normal Activity ............................................ 30

1.2 Estimating the Effort ......................................................................... 30
  1.2.1 The Technical Upgrade .......................................................... 30
  1.2.2 The Number of SAP Objects Modified .................................... 31
  1.2.3 Custom Developments .......................................................... 34
  1.2.4 ABAP Unicode Syntax Requirements .................................... 37
  1.2.5 Obsolete Transactions .......................................................... 37
  1.2.6 Obsolete Custom-Developed Programs ................................. 39
  1.2.7 Estimating the Functional Effort ........................................... 39
  1.2.8 Business Example: Upgrade from ECC 5.0 to ECC 6.0 EHP 7 ............................................................... 39
  1.2.9 Estimating the Technical Upgrade Runtime for the Production System ............................................................... 41

1.3 Pre-Upgrade Considerations ............................................................. 42
  1.3.1 What Is the Level of Effort? .................................................. 42
  1.3.2 Which Release to Choose? .................................................... 42
  1.3.3 Where to Start? ................................................................... 43

1.4 Factors Influencing the Complexity of an Upgrade ......................... 43
  1.4.1 Technology-Related Factors ............................................... 44
  1.4.2 Project-Related Factors ....................................................... 59
  1.4.3 Business-Related Factors .................................................... 63
  1.4.4 Forgotten Factors ............................................................... 64

1.5 The Project Team .............................................................................. 65
  1.5.1 Project Management Team .................................................. 65
  1.5.2 Technical Upgrade Team ..................................................... 66
  1.5.3 Functional Work Groups ..................................................... 67
## 4 A Guided Tour of the Upgrade Tools ........................................ 167

4.1 Obtaining the SUM ................................................................ 168
4.1.1 Download the Software ........................................ 168
4.1.2 Download the Documentation .................................. 169
4.1.3 The SUM Note ..................................................... 171
4.1.4 Other Guides and Notes ......................................... 173
4.2 Setting up the SUM .......................................................... 175
4.2.1 Planning Disk Space ............................................... 175
4.2.2 Software on the Server .......................................... 178
4.2.3 Software on the Workstation ................................... 178
4.2.4 Extraction ............................................................. 179
4.2.5 SUM Roles: Administrator and Observer .............. 180
4.2.6 Giving the SUM a Trial Run .................................... 181
4.2.7 Connection between SAP Server and Workstation .... 189
4.3 Using the SUM ............................................................... 193
4.3.1 The File Menu ...................................................... 193
4.3.2 The User Menu ..................................................... 194
4.3.3 The Alert Menu ..................................................... 195
4.3.4 The Update Menu .................................................. 196
4.3.5 The ABAP Menu .................................................... 197
4.3.6 The Java Menu ..................................................... 198
4.3.7 The Help Function ................................................ 199
4.4 Special Features of the SUM ............................................ 200
4.4.1 Setting Breakpoints (ABAP) ................................. 200
4.4.2 Setting Breakpoints (Java) ...................................... 202
4.4.3 Reinitializing the Administrator password ............. 203
4.4.4 Starting the GUI on Nondefault Ports ................. 203
4.5 Summary ................................................................. 204

## 5 Upgrading the ABAP System .................................................. 205

5.1 Planning the ABAP Upgrade ........................................... 207
5.2 ABAP Upgrade Timeline ............................................... 209
5.2.1 Downtime before the Upgrade ............................... 209
5.2.2 System Availability during the ABAP Upgrade .......... 210
5.2.3 System Resources and Upgrade Scenarios .............. 212
5.2.4 Near-Zero Downtime Maintenance (nZDM) ............ 214
5.2.5 Backups ............................................................. 215
5.2.6 Downtime after Go-Live ........................................ 217
5.2.7 Time Schedule during the Technical Upgrade .......... 218
5.3 The Shadow Repository and Shadow Instance ............... 220
5.4 The ABAP Upgrade Directory ...................................... 220
5.5 SAUP ........................................................................ 223
5.6 Starting the ABAP Upgrade ........................................... 228
5.6.1 Prerequisites ......................................................... 228
5.6.2 Software Update Manager Roadmap Steps ............. 232
5.7 Initialization Roadmap Step .......................................... 232
5.8 Extraction Roadmap Step .............................................. 239
5.9 Configuration Roadmap Step ........................................ 244
5.9.1 Upgrade Strategy .................................................. 244
5.9.2 Configuring the Downtime-Minimized Strategy ....... 246
5.9.3 Package Inclusion ................................................ 250
5.9.4 Patch Binding ...................................................... 253
5.9.5 Customer Transport ............................................. 254
5.9.6 Requests for SPDD and SPAU ......................... 255
5.9.7 Shadow Instance ................................................ 257
5.9.8 List of Upgrade Files ............................................ 259
5.10 Checks Roadmap Step .................................................. 259
5.10.1 Saving Variants ................................................... 259
5.10.2 The ASU Toolbox ............................................... 260
5.11 Preprocessing Roadmap Step ....................................... 263
5.11.1 Error Stop for Update Requests ......................... 264
5.11.2 Development and Transport Lock ....................... 265
5.11.3 Unattended Run to Modification Adjustment Stop .... 267
5.11.4 Stop for Modification Adjustment ....................... 267
5.11.5 The Activation Phase ACT_UPG .......................... 270
5.11.6 Activation Errors ............................................... 272
5.11.7 Incremental Conversion (ICNV) ......................... 277
5.11.8 Remaining Phases until Downtime ..................... 284
5.11.9 Actions before Entering Downtime ..................... 284
5.11.10 Start of Downtime ............................................ 288
5.11.11 Backup Database and SUM Directory .............. 289
5.11.12 Disable Database Archiving ............................. 290
5.11.13 Restart the Upgrade ......................................... 290
5.12 Execution Roadmap Step ............................................. 291
5.12.1 Logging On to SAP during Downtime .................. 292
5.12.2 Unlock the System to Correct Errors ................... 292
5.12.3 The Switch Phases: EU_SWITCH and KX_SWITCH_1 .... 293
5.12.4 Table Conversion Phase PARCONV_UPG .......... 293
5.12.5 Control Data Import: TABIM_UPG ...................... 293
5.12.6 The XPRAS Phases .............................................. 294
5.12.7 Time Schedule during the Technical Upgrade ........ 218
5.12.8 Downtime after Go-Live ...................................... 217
5.12.9 Connection between SAP Server and Workstation ... 189
5.12.10 Special Features of the SUM ......................... 200
5.12.11 Summary ....................................................... 204

## Contents

- **4 A Guided Tour of the Upgrade Tools**
  - 4.1 Obtaining the SUM
  - 4.2 Setting up the SUM
  - 4.3 Using the SUM
  - 4.4 Special Features of the SUM
  - 4.5 Summary

- **5 Upgrading the ABAP System**
  - 5.1 Planning the ABAP Upgrade
  - 5.2 ABAP Upgrade Timeline
  - 5.3 The Shadow Repository and Shadow Instance
  - 5.4 The ABAP Upgrade Directory
  - 5.5 SAUP
  - 5.6 Starting the ABAP Upgrade
  - 5.7 Initialization Roadmap Step
  - 5.8 Extraction Roadmap Step
  - 5.9 Configuration Roadmap Step
  - 5.10 Checks Roadmap Step
  - 5.11 Preprocessing Roadmap Step
  - 5.12 Execution Roadmap Step

## Page Numbers

- 4.1 Obtaining the SUM: 168
- 4.2 Setting up the SUM: 175
- 4.3 Using the SUM: 193
- 4.4 Special Features of the SUM: 200
- 4.5 Summary: 204
- 5.1 Planning the ABAP Upgrade: 207
- 5.2 ABAP Upgrade Timeline: 209
- 5.3 The Shadow Repository and Shadow Instance: 220
- 5.4 The ABAP Upgrade Directory: 220
- 5.5 SAUP: 223
- 5.6 Starting the ABAP Upgrade: 228
- 5.7 Initialization Roadmap Step: 232
- 5.8 Extraction Roadmap Step: 239
- 5.9 Configuration Roadmap Step: 244
- 5.10 Checks Roadmap Step: 259
- 5.11 Preprocessing Roadmap Step: 263
- 5.12 Execution Roadmap Step: 291

## Additional Notes

- 4.4.1 Setting Breakpoints (ABAP)
- 4.4.2 Setting Breakpoints (Java)
- 4.4.3 Reinitializing the Administrator password
- 4.4.4 Starting the GUI on Nondefault Ports
- 5.9.1 Upgrade Strategy
- 5.9.2 Configuring the Downtime-Minimized Strategy
- 5.9.3 Package Inclusion
- 5.9.4 Patch Binding
- 5.9.5 Customer Transport
- 5.9.6 Requests for SPDD and SPAU
- 5.9.7 Shadow Instance
- 5.9.8 List of Upgrade Files
- 5.10.1 Saving Variants
- 5.11.1 Error Stop for Update Requests
- 5.11.2 Development and Transport Lock
- 5.11.3 Unattended Run to Modification Adjustment Stop
- 5.11.4 Stop for Modification Adjustment
- 5.11.5 The Activation Phase ACT_UPG
- 5.11.6 Activation Errors
- 5.11.7 Incremental Conversion (ICNV)
- 5.11.8 Remaining Phases until Downtime
- 5.11.9 Actions before Entering Downtime
- 5.11.10 Start of Downtime
- 5.11.11 Backup Database and SUM Directory
- 5.11.12 Disable Database Archiving
- 5.11.13 Restart the Upgrade
- 5.12.1 Logging On to SAP during Downtime
- 5.12.2 Unlock the System to Correct Errors
- 5.12.3 The Switch Phases: EU_SWITCH and KX_SWITCH_1
- 5.12.4 Table Conversion Phase PARCONV_UPG
- 5.12.5 Control Data Import: TABIM_UPG
- 5.12.6 The XPRAS Phases
Contents

8.3.4 Checking Inconsistent InfoObjects .................................. 395
8.3.5 Converting Data Classes of InfoCubes ......................... 397
8.3.6 Migrating InfoPackage Groups to Process Chains ......... 399
8.3.7 Migrating to the New Reporting Authorization Concept ..... 400
8.3.8 Checking for Incompatibilities with Source Release SAP BW 3.5 ............................................................... 400
8.3.9 Creating and Running Reports for Open Hub ................ 401
8.3.10 Applying Corrections to Prevent the Loss of Function Groups ................................................................. 401
8.3.11 Checking for Discontinued Query Features ................ 401
8.3.12 Preparing the System for Change Data Type for Characteristics ................................................................. 402
8.3.13 Executing Automated Housekeeping Tasks ................ 403
8.3.14 Executing Automated Before-Upgrade Tasks ............... 404
8.4 The Preparation and Upgrade Process for SAP Business Warehouse ................................................................. 406
8.4.1 During Preparation .......................................................... 406
8.4.2 During the Upgrade .......................................................... 408
8.5 Upgrade Postprocessing for SAP Business Warehouse ...... 409
8.5.1 Overview of SAP BW Postprocessing Actions ............... 409
8.5.2 Installing the Java Components ...................................... 410
8.5.3 Postprocessing Activities for BI Java .............................. 410
8.6 Summary ............................................................................. 412

9 Upgrading SAP SCM ................................................................. 413
9.1 SAP APO Components ....................................................... 414
9.1.1 Upgrade of liveCache and Optimizer ............................ 415
9.1.2 Integration with SAP ERP ............................................... 415
9.2 SAP SCM Architecture .......................................................... 416
9.2.1 Platform Support for liveCache and Optimizer ............. 416
9.2.2 Virtualization ................................................................. 418
9.3 Dependencies between SAP SCM and Backend Systems ................................................................. 419
9.4 Functional Aspects during the Technical Upgrade .......... 420
9.5 Technical Upgrade Preparation, Documentation, and Media ................................................................. 421
9.5.1 Additional Documents and Notes ................................. 422
9.5.2 Additional Upgrade Media ............................................. 422
9.6 The Technical Upgrade Process for SAP SCM .................. 423
9.7 /SAPAPO/OM_LC_UPGRADE_70 Section A ....................... 427
9.7.1 Delete Downloaded Table and Logs (Action A1) .......... 428
9.7.2 Delete Superfluous Planning Versions (A2) ................. 428
9.7.3 Consistency Check (A3) ............................................... 428
9.7.4 Consistency Check for Activities Data (A4) ................. 429
9.7.5 liveCache/LCA Build Checks (A5) ............................... 429
9.7.6 Consistency Checks for DP/SNP Time Series (A6) ....... 429
9.7.7 Consistency Check for Time Streams (A7) .................... 430
9.7.8 Resume the Upgrade .................................................... 430
9.8 Entering Downtime .............................................................. 431
9.9 /SAPAPO/OM_LC_UPGRADE_70 Section B ....................... 431
9.9.1 Repeat Consistency Checks .......................................... 431
9.9.2 Check Space in the Database ...................................... 432
9.9.3 Download liveCache Data (B1) .................................... 432
9.9.4 Stop liveCache (B2) ..................................................... 433
9.9.5 Complete Backup (B3) ............................................... 433
9.10 Upgrade Downtime Phases ................................................. 433
9.11 liveCache Upgrade ............................................................. 434
9.11.1 Upgrade the Management GUI ................................... 435
9.11.2 Obtain User Access on the liveCache Server ............. 435
9.11.3 Stop Other liveCache and SAP MaxDB Instances on the Server ................................................................. 436
9.11.4 Execute the Upgrade Script ........................................ 436
9.11.5 Install the liveCache Client Software ........................ 437
9.11.6 Start/Stop Test .......................................................... 438
9.11.7 Adapt liveCache Parameters ...................................... 438
9.11.8 Parameter Check ........................................................ 439
9.11.9 Final Actions ............................................................. 441
9.11.10 Start CIF Queues (C10) ............................................. 448
9.12 Optimizer Upgrade ............................................................ 441
9.13 /SAPAPO/OM_LC_UPGRADE_70 Section C ....................... 445
9.13.1 Maintain Logical Database Connection (C1) ............... 446
9.13.2 Refresh Database Statistics (C2) ............................... 446
9.13.3 Load Master Data (C3) ............................................... 446
9.13.4 Upload liveCache Data (C4) ........................................ 446
9.13.5 Compare liveCache with Download (C5) .................... 447
9.13.6 Convert Transport Requests (C6) ............................... 447
9.13.7 liveCache/LCA Build Checks (C7) ............................. 447
9.13.8 Activate Logging (C8) ............................................... 447
9.13.9 Complete Backup (C9) ............................................. 448
9.13.10 Start CIF Queues (C10) ............................................. 448
9.14 Final Upgrade Phases ........................................................ 448
9.15 Prepare for Return to Production ....................................... 448
9.16 /SAPAPO/OM_LC_UPGRADE_70 Section D ....................... 449
9.17 Summary ............................................................................. 449
## 10 Upgrading SAP CRM

10.1 SAP CRM Architecture ................................................................. 451
10.1.1 SAP CRM Server ................................................................. 451
10.1.2 SAP CRM Mobile Client Component .................................... 454
10.1.3 SAP NetWeaver Search and Classification ............................... 455
10.2 SAP CRM Specific Tasks ............................................................ 455
10.2.1 Connections to Backend Systems ........................................... 455
10.2.2 Dependency of SAP CRM on Other SAP Applications ............ 456
10.2.3 SAP CRM Java Components ................................................... 456
10.3 The Preparation and Upgrade Process for SAP CRM .................... 457
10.3.1 During Preparation ............................................................... 457
10.3.2 During the Upgrade .............................................................. 457
10.4 Upgrade Postprocessing for SAP CRM ....................................... 462
10.4.1 Internet Pricing and Configurator ......................................... 462
10.4.2 Follow-Up Activities for the Middleware .............................. 463
10.4.3 Reregistering Inbound Queues ............................................. 465
10.4.4 Rereleasing Replication and Realignment Queues ................ 465
10.4.5 Other Upgrade Postprocessing Steps .................................... 465
10.4.6 SAP CRM Java Components ............................................... 466
10.5 Upgrading SAP CRM Mobile Client Components ...................... 467
10.6 Summary ...................................................................................... 468

## 11 Upgrading SAP SRM

11.1 SAP SRM Architecture ............................................................... 469
11.1.1 SAP SRM Server ................................................................. 470
11.1.2 SAP ERP ............................................................................ 471
11.1.3 SAP BW ................................................................. 472
11.1.4 SAP NetWeaver Search and Classification ............................... 472
11.1.5 SAP NetWeaver Master Data Management ............................ 472
11.2 Specific Tasks for SAP SRM ........................................................ 473
11.2.1 Connections to Backend Systems ........................................... 473
11.2.2 Dependency of SAP SRM on Other SAP Applications ......... 473
11.2.3 Preparation ................................................................. 475
11.2.4 During the Upgrade ........................................................... 475
11.2.5 Upgrade Postprocessing ....................................................... 478
11.2.6 SAP SRM Java Components ............................................... 481
11.3 Summary ...................................................................................... 483

## 12 Upgrading the SAP Enterprise Portal

12.1 SAP Enterprise Portal Architecture ......................................... 485
12.1.1 SAP Enterprise Portal Add-Ons ............................................. 487
12.1.2 SAP Enterprise Portal Services ............................................. 488
12.1.3 SAP Enterprise Portal Standalone Engines ............................. 489
12.2 SAP Enterprise Portal Upgrade Approach .................................. 490
12.3 Upgrade Preparation and Process .............................................. 492
12.3.1 Dual-Stack Split ............................................................... 492
12.3.2 Revert to Default Desktop .................................................. 492
12.3.3 The Universal Worklist ....................................................... 493
12.3.4 Wiki and Forums ............................................................... 493
12.3.5 Knowledge Management and Collaboration ....................... 494
12.4 Upgrade Postprocessing for the SAP Enterprise Portal ............. 495
12.4.1 The Universal Worklist ....................................................... 495
12.4.2 Knowledge Management and Collaboration ....................... 495
12.4.3 Migrating Portal Applications ............................................. 496
12.5 Summary ...................................................................................... 496

## 13 Upgrading SAP Process Integration and SAP Process Orchestration

13.1 SAP Process Integration versus SAP Process Orchestration .......... 498
13.2 In-Place Upgrade versus Side-by-Side Migration ....................... 500
13.2.1 Decision Factors ............................................................... 500
13.2.2 Advantages/Disadvantages ................................................ 502
13.3 Side-by-Side Deployment ........................................................ 504
13.3.1 Migration Plan ............................................................... 504
13.3.2 Directory Content Migration Tool ....................................... 505
13.4 In-Place Upgrade ................................................................. 507
13.4.1 Preparation Activities ........................................................ 507
13.4.2 Follow-Up Activities ........................................................ 510
13.5 Summary ...................................................................................... 513

## 14 Upgrading SAP Solution Manager

14.1 SAP Solution Manager 7.1 Infrastructure ................................... 515
14.1.1 System Landscape Database ................................................. 516
14.1.2 Landscape Management Database ....................................... 517
14.1.3 SAP Host and Diagnostic Agents ......................................... 518
14.1.4 CA Wily Introscope ........................................................... 518
14.2 Upgrade Approach ................................................................. 518
   14.2.1 System Architecture ....................................................... 518
   14.2.2 Upgrade versus New Installation ................................. 519
   14.2.3 The Upgrade Planning Tool ............................................ 519
   14.2.4 The SAP Landscape during the Upgrade ....................... 522
14.3 The Preparation and Upgrade Process for SAP Solution Manager ... 523
   14.3.1 How to Upgrade SAP Solution Manager ......................... 523
   14.3.2 Updating the SLD Content to the Latest Version .......... 523
   14.3.3 Upgrade to SAP Solution Manager 7.1 ......................... 524
   14.3.4 Set End-to-End Root-Cause Analysis in
       Maintenance Mode ........................................................ 524
   14.3.5 Upgrade CA Wily Introscope Enterprise Manager ......... 527
   14.3.6 Upgrade the SAP Host and Diagnostics Agents on
       All Managed Systems .................................................. 527
   14.3.7 SAP Solution Manager Add-Ons ................................. 529
14.4 Post-Upgrade Activities ...................................................... 530
   14.4.1 Perform Delta Configuration of SAP Solution
       Manager and Managed Systems ..................................... 530
   14.4.2 Repeat Managed Systems Configuration
       (Only for SAP PI/PO Systems) ...................................... 531
14.5 Summary ............................................................................. 532

Appendices....................................................................................... 533
A SAP Releases and Upgrade Paths .................................................. 535
   A.1 SAP NetWeaver and SAP Basis Releases ......................... 535
   A.2 SAP ERP Releases ....................................................... 537
B Database Transaction Log Modes .............................................. 541
C SAP Notes ............................................................................... 543
D References .............................................................................. 549
   D.1 SAP Service Marketplace ................................................. 549
   D.2 SAP Installation and Documentation Manuals .................. 549
   D.3 SAP NetWeaver How-To Guides ....................................... 551
E SAP NetWeaver Search and Classification (TREX) ....................... 553
F Single Code Pages ................................................................. 555
G The Authors ............................................................................. 557

Index.......................................................................................... 559
Index

Enhancement package, 100, 158, 537
advantages, 101
process in upgrade, 251
version numbering, 97
Enterprise Extensions (EA), 96, 537
EP component (Enterprise Portal), 485
EP Core component (Enterprise Portal Core), 485
EPS Inbox, 176
Evaluation, 300, 361
feedback to SAP, 306
Extended Configuration Management (XCM), 466
Extended maintenance, 28
Extension index, 375
Extension set, 537
External consultants, 67
Externally managed projects, 70
Extraction queues, 264

INDEX

Hardware capacity, 41, 142, 144, 206
High availability, 288
setup, 502
Hostname resolution, 191
How-To Guides, 551

J

ICF -> see Internet Communication Setup
ICNV, 103, 277
Inbound Queue Monitor, 459, 476
Incremental Table Conversion (ICNV), 103, 106, 213, 277
at start of downtime, 286
configuration, 249
run, 281
select tables, 280
switch during downtime, 293
switch function, 284

K

J2EE_ADMIN, 333
Java
backup, 362
components media, 157
disk space, 328
passwords, 333
preparation guide, 328
shadow instance, 330
shadow schema, 353
sizing requirements, 145
Java Connector (JCo), 475, 482
Java Virtual Machine (JVM), 331
Java Web Start, 178
troubleshooting, 189
JNLP file, 190

L

LAN Check by Ping, 52
Live Auction Cockpit (LAC), 471, 474

catalog, 483
migration of data, 481
migration of resource customizations, 483
SAP SRM 7.01, 474
test after upgrade, 482
liveCache, 414, 429
client software, 437
collection after upgrade, 447
database statistics, 446
download data, 432
download table, 428, 432, 449
for SAP CRM, 452
logging, 447
logical database connections, 446
master data, 446
media for upgrade, 422
parameter check, 439
parameters, 438
save data to backup table, 431
start/control upgrade, 436
supported platforms, 416
upgrade, 415, 434
upgrade guide, 422
upgrade script, 436
upload data, 446
virtualization, 418
liveCache Applications (LCA), 429, 447
for SAP CRM, 452
liveCache/LCA Build Checks, 447
LMDB, 158, 517
Lock development, 265
Lock users, 287
Logical system name
SAP BW, 393
LONGPOST.LOG, 303, 312

M

Maintenance, 28
transaction, 159
Maintenance Optimizer (MOP), 131, 144, 158, 159, 515
SAP NetWeaver Master Data Management Catalog, 473
SAP Solution Manager, 524
Wiki and Forum applications, 487

F

FINDSTR, 221
Firewall, 192
Frontend software, 54
upgrade, 116
FTE (Full-Time Equivalent), 33
FTE (Full-time Equivalent), 36, 39, 75
Functional downtime, 212
Functional effort, 39
Functional work groups, 67

G

GENSTATUS, 464, 479
Global Status Reporting, 80
Glossary and terminology data, 312
GN_CDBINDEX, 464
GN_START, 464
Go live, 217
Go/no-go decision, 79
grep, 221

H

HANA -> see SAP HANA
Hardware
for upgrade, 50, 105

I

ICF -> see Internet Communication Setup
ICNV, 103, 277
Inbound Queue Monitor, 459, 476
Incremental Table Conversion (ICNV), 103, 106, 213, 277
at start of downtime, 286
configuration, 249
run, 281
select tables, 280
switch during downtime, 293
switch function, 284
Index, 375
Industry Solutions (IS), 96
InfoCube, 390
activation, 396
data class, 397
InfoObjects
consistency check, 395, 406
InfoPackage
migrate to process chains, 399
Information Broadcasting, 410
Informix, 148
In-memory database management system, 414
Installed notes, 31
Instance directory, 147, 329
Interfaces, 53
stopping, 287
Internet Communication Framework (ICF), 453
Internet Communication Manager (ICM), 94
Internet Pricing and Configurator (IPC), 453, 462, 471, 474, 480
Internet Transaction Server (ITS)
in Upgrade Dependency Analyzer, 48
IPC, 453
Isolating the system, 285
ITS, 48
views, 453, 492

J

J2EE_ADMIN, 333
Java
backup, 362
catalog, 157
disk space, 328
passwords, 333
preparation guide, 328
shadow instance, 330
shadow schema, 353
sizing requirements, 145
Java Connector (JCo), 475, 482
Java Virtual Machine (JVM), 331
Java Web Start, 178
troubleshooting, 189
JNLP file, 190
JVM, 331

K

Kernel, 143
Kernel DVD, 156
Key users, 67
Keyword (for update), 173, 238
KMC, 488
Knowledge Management and Collaboration (KMC), 488, 493
API changes, 494
wrify deployment, 495

L

LAN Check by Ping, 52
Landscape
for upgrade, 55, 109
prerequisites, 110
setup options, 110
Landscape Management Database (LMDB), 158, 317, 516, 517
Languages, 312, 555
disk space, 150
impact on upgrade runtime, 106
supplementation, 312
LC10, 433, 448
LCA -> see liveCache Applications

M

Maintenance, 28
transaction, 159
Maintenance Optimizer (MOP), 131, 144, 158, 159, 515
SAP NetWeaver Master Data Management Catalog, 473
SAP Solution Manager, 524
Wiki and Forum applications, 487
Index

Management GUI, 435
Management involvement, 63
Master guide, 126, 174
Master project plan, 71, 74
MaxDB Database Studio, 423, 435
MCO, 108
MDM, 472
MDMP (Multiple Display Multiple Processing), 29, 43
Media, 154
directory, 147, 228
download path, 155
list, 130, 174
selecting, 155
Memory
for Java upgrade, 330
Microsoft Management Console (MMC), 288
MMC, 288
MMR_CNTL (table), 464
Mobile Client Message Recovery, 464
Modification Adjustment, 31, 33, 69, 101, 208
Mobile Client Message Recovery, 464
Modification Browser, 31, 369
Modifications, 44
modified objects, 31
Multiple Components in One Database (MCOD), 108
upgrades, 122
MultiProvider, 410
 MW_CHECK, 465, 479
mySAP ERP, 536
Near-Zero Downtime Management (nZDM), 105, 214, 330
configuration, 246
New functionality, 61
Number ranges, 394

O
Obsolete custom programs, 39
Obsolete transactions, 37
ODS, 390
objects, 407
Operating system
maintenance life, 28
Operation mode, 287
Optimizer, 414
installation, 423
installation guide, 422
media for installation, 423
supported platforms, 416
update statistics, 308
upgrade, 415, 441
virtualization, 418
oraroot.sh, 308
OS/DB Migration, 148
Outbound Queue Monitor, 476
Outdated SAP version, 28

P
PAM, 416
Parallelism, 105
Password
change, 194
for SUM roles, 186
Java, 333, 343
Java Secure Store, 351
reset SUM password, 203
Path name, 149
change, 150, 153
OS independent, 180
People-Centric UI, 454
Performance and tuning, 52
Performance testing, 82
PFCG, 318
PI (Process Infrastructure), 93
PI_BASIS, 420
Planning
levels, 71
versions, 428

Q
QRFC monitor, 478
Quality assurance system, 111
Queries that will not run correctly, 402

R
R/3 Enterprise, 536
R/3 plug-in compatibility, 49
R&R Queues, 461
RJAR2, 459
RJAR4, 459
Rload, 247
fractional number of imports, 247
Rtrans, 230, 247
Ramp-up project, 42
RAR files, 157
Reasons for upgrading, 27
Regression test, 82, 85, 101
Release
choice of target, 42
requirements, 144
strategy, 44, 98
Release Information Note (RIN), 164
RemoteCube, 390
Repairs, 286
Replication and realignment queues, 465
Repository objects, 31
Repository switch upgrade, 102
Reset upgrade, 197, 323
Resource comparison notes, 144
Resource-minimized upgrade, 104, 212
for SAP BW, 408
single system configuration, 244
RFC server groups, 310
RIN, 164
Roadmap step, 232
Checks, 259
Checks (Java), 352
Configuration, 244
Configuration (Java), 343
Execution, 291
Execution (Java), 358
Extraction, 239
Extraction (Java), 342
Initialization, 232
Initialization (Java), 336
Postprocessing and Finalization, 299
Postprocessing and Finalization (Java), 359
Preprocessing, 263
Preprocessing (Java), 353
Roles, 317
Root access (Unix, Linux), 229, 307, 362
Root-cause analysis, 524
RSA1, 400
RSD1, 395, 407
RSDCUBE, 396
RSRV, 395
RSUPGRCHECK, 396
RSXPRRAUP, 296
R204, 287
R210, 218, 309
R212, 310
R270, 317
SAINT, 530
check installed version, 241
updating, 231
Sales order processing, 452
SAMT, 35
Sandbox system, 57, 110, 111
interfaces, 53
SAP Accelerated Value Assessment, 141
SAP APO, 414
development step time, 266
integration with SAP ERP, 415
SAP Application Components, 154
SAP Basis, 98
effort, 40
SAP Basis Plug-In, 391, 420
SAP BI
SAP BW component, 390
SAP Bidding Engine, 470
SAP Business Connector
in Upgrade Dependency Analyzer, 48
SAP Business Explorer (BEx), 391
SAP Business Information Warehouse (SAP BW), 535
SAP Business Process Management (BPM), 93, 498
SAP Business Rules Management (BRM), 498
SAP Business Suite
upgrade/update guide, 129
SAP Business Warehouse (SAP BW), 93
SAP BusinessObjects BI
configuration wizard, 411
connections, 411
Java components, 410
SAP BW, 389
add-ons, 390
authorization concept, 400
before-upgrade tasks, 404
BW 3.5 as source version, 400
twists to objects, 408
characteristics data type, 402
development step time, 266
discontinued query features, 401
housekeeping tasks, 403
in APO systems, 414
number ranges, 394
on SAP HANA database, 118
Open Hub service, 401
SAP Collections Management, 452
SAP Community Network (SCN), 146
SAP Content Server
in Upgrade Dependency Analyzer, 48
SAP CRM
add-ons, 452
architecture, 451
archive messages, 460
backend system, 455
business function prerequisites, 456
CDB tables, 464
CPRERPM, 452
CRM Core, 452
CRM server component, 451
dependencies, 456
FINBASIS, 452
Java components, 452, 456, 466
liveCache Applications, 452
middleware, 463, 464
Mobile Application Studio, 455
Mobile Client Component, 454, 467
mobile client messages, 464
Mobile Components, 452
mobile system landscape, 454
orkgroup client, 467
postprocessing, 465
preparation phases, 457
queues (inbound), 460, 465
queues (replication and realignment), 461, 465
TREX, 455
Web Channel, 452
Web Channel scenarios, 451
WebClient User Interface, 543
WFMCORE, 452
workgroup server, 467
SAP CRM 4.0, 463, 466
SAP CRM Adapter Framework, 466
SAP Dispute Management, 452
SAP E-Commerce
in CRM upgrade, 467
SAP End-User Delta Training, 141
SAP Enterprise Portal, 92
add-ons, 487
architecture, 485
Component Monitor, 494
default desktop, 492
Enterprise Workspaces, 487
SAP Enterprise Portal (Cont.)
Forums, 487, 493
Knowledge Management and Collaboration, 488
migrate portal applications, 496
postprocessing activities, 495
standard clones, 489
Universal Worklist (UWL), 493, 495
upgrade vs. migration, 491
Wiki, 487, 493
SAP ERP, 95, 99, 455, 536
releases, 537
SAP ERP Central Component (ECC), 96, 537
SAP Exchange Infrastructure (XI), 497
SAP Financial Supply Chain Management (FSCM), 452
SAP Functional Upgrade Service, 53
SAP GoingLive Functional Upgrade Check, 137, 140
SAP HANA, 29, 94, 118
and liveCache, 414
SAP Host Agent, 528
SAP NetWeaver architecture, 91
upgrade/update guide, 129
versions, 535
SAP NetWeaver 2004, 332
SAP NetWeaver Application Server, 93
SAP NetWeaver Developer Studio (NWDS), 496
SAP NetWeaver Master Data Management (MDM), 472
Catalog, 473
SAP Notes
list of, 543
SAP PI Adapter Framework, 508
SAP Plug-In, 415, 420, 456, 473
SAP Process Integration, 93
dual-stack system, 118
SAP Process Integration (PI), 497
migrate to SAP PO, 504
SAP Process Orchestration (PO), 93
Adapter Engine, 499
Adapter Framework, 508
background jobs, 512
configuration wizard, 510
Directory Content Migration Tool, 505
SAP Process Orchestration (PO) (Cont.)
in place upgrade, 502, 507
PO vs. PI, 498
release restriction for other systems, 504
side-by-side deployment, 503, 504
upgrade vs. migration, 500
SAP Quick Sizer, 51, 146
SAP R/3
Enterprise, 99, 537
Plug-In, 393
releases, 537
SAP releases, 95, 535
SAP Safeguarding for Upgrade, 137, 140
SAP SCM architecture, 416
SAP SCM Server, 138
SAP Solution Manager, 56, 536
ABAP upgrade configuration, 317
Add-Ons, 529
applications operations guide, 130
architecture, 518
as source of upgrade information, 143
automate test scenarios, 87
Change Request Management (Ch aRM), 521
delta configuration, 530
diagnostic agent, 527
diagnostic agents, 518
host agent, 518, 527
in Upgrade Dependency Analyzer, 48
managed systems configuration (for SAP PI/PO), 531
MOPZ, 131
root-cause analysis, 518, 524
stack XML file, 524
synchronized landscape information, 158
Upgrade Planning Tool, 519
Upgrade Roadmap, 88
upgrade vs. install, 519
SAP SRM
architecture, 469
backed connections, 473
dependencies, 473
deregistered queues, 477
Java components, 481
middleware, 479
queues (inbound), 477, 479
registering as local system, 480
Spend Analysis, 472
SRM Server component, 470
user management, 482
Secure Store (Java), 333, 351

SAP Support, 325

SAP Support Portal, 131

SAP Support Services, 96

SAP Test Management Optimization service, 88

SAP Upgrade Hosting, 141

SAP Upgrade Weekend Support, 141

SAP Value Assessment, 141

SAP Web Application Server, 29, 536

SAP, ALL profile, 229

SAP_NEW profile, 317

SAPA* activation logs, 271

SAPCAR, 290

SAPGUI, 454

SAPJVM, 178, 331

SAProuter, 288

SAProuter.sh, 308

SAPOSCOL, 288

SAPOSCOL, 288

SE95_UTIL, 377

SE95, 31

SE38, 37

SE11, 33

SE16, 35

SE37, 36

SE38, 37

SE95, 31

SE95_UTIL, 377

Software Update Manager (SUM), 101, 167

ABAP breakpoints, 200

accept non-severe errors, 323

alerts, 195

Application-Specific Upgrade Toolbox, 116

backup directory, 215
categories, 170

central note, 171
core SAP Notes, 131
directory, 175
download, 168

error stop, 321

existing software prerequisites, 178

expert mode, 245

ignore error, 322

manualy prepared directory, 235

network ports, 192, 203

passwords, 230

reset administrator password, 203

roadmap steps, 232

roles, 180

run server on upgrade host, 175

trial run, 181

update process, 196

URL, 184

user guides, 169

user name, 166

SOLMAN_SETUP, 523, 530

Solution agent, 101

Solution Manager, 523, 530

Solution gap, 28

Solution Manager → see SAP Solution Manager

Solution monitoring, 88

SPAM

check installed version, 241

upgrading, 231

SPAU, 31, 33, 206, 209, 384, 387

dictionary objects handled by, 366

SAP CRM, 463

specify transport, 255

transport, 33

SPAU, ENH, 365, 387

SPDD, 206, 365

append structure, 377

custom fields, 377

data elements, 382

domains, 382

enable development changes, 368

field format changes, 381

SPDD (Cont.)

indexes, 382

log on to shadow instance, 368

main steps, 269

object list, 370

proposal, 377

run, 267

specify transport, 255

table technical settings, 382

transport, 386

SSM2, 409

ST02, 52, 320

monitor, 217

ST03

transaction profiles, 36

ST03N, 52

ST06, 52

ST13, 529

ST14, 529

ST22, 320

ST-A/PI, 251, 529

Stack XML file, 89, 131, 164, 188, 336

before upgrade, 228

tenter path for upgrade, 234

ersors, 348

for Solution Manager upgrade, 524

Stack-independent files, 163

STAD, 52

Staffing, 67

Standard downtime upgrade strategy, 104

STARTUP script, 181

Status reporting, 80

STC01, 405

Step-by-step plan, 72

STMS, 311, 316

ST-PI, 96, 529

Structure, 268

SU10, 287, 319

SU25, 317

SUM directory

backup, 289

SUM GUI

features, 193

ports, 203

start, 184

stop the server, 188

troubleshoot, 193

SAP Strategic Enterprise Management (SEM), 391

SAP Support, 325

SAP Support Portal, 131

SAP Test Management Optimization service, 88

SAP Upgrade Hosting, 141

SAP Upgrade Weekend Support, 141

SAP Value Assessment, 141

SAP Web Application Server, 29, 536

SAP, ALL profile, 229

SAP_NEW profile, 317

SAPA* activation logs, 271

SAPCAR, 290

SAPGUI, 454

SAProuter, 288

SAProuter.sh, 308

SAPOSCOL, 288

SAPup, 168, 223

SAPUI5, 92

SAPSetup, 116

SAProuter, 288

SAPupConsole.log, 222, 223

SAPup.log, 222

SCAN_DOWNLOADDIR, 240

SCC, 319

SCN → see SAP Community Network

SCS instance, 344

SCM, 332

sdirectory, 152

SE06, 319

SE09, 286, 386

SE11, 33

SE16, 35

SE37, 36

SE38, 37

SE95, 31

SE95_UTIL, 377
<table>
<thead>
<tr>
<th>Table</th>
<th>technical settings, 382</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Browser, 35</td>
</tr>
<tr>
<td>Table</td>
<td>DDATT, 398</td>
</tr>
<tr>
<td>Table</td>
<td>sizes</td>
</tr>
<tr>
<td>Impact on upgrade runtime, 106</td>
<td></td>
</tr>
<tr>
<td>Tables</td>
<td>spaces</td>
</tr>
<tr>
<td>drop after upgrade, 309</td>
<td></td>
</tr>
<tr>
<td>MCOO, 122</td>
<td></td>
</tr>
<tr>
<td>Technical upgrade, 30</td>
<td></td>
</tr>
<tr>
<td>analyst, 66, 68</td>
<td></td>
</tr>
<tr>
<td>process, 208</td>
<td></td>
</tr>
<tr>
<td>team, 66</td>
<td></td>
</tr>
<tr>
<td>time schedule, 218</td>
<td></td>
</tr>
<tr>
<td>Test Cockpit, 80</td>
<td></td>
</tr>
<tr>
<td>Test cycles, 84</td>
<td></td>
</tr>
<tr>
<td>Testing, 56, 69, 81, 216</td>
<td></td>
</tr>
<tr>
<td>automated, 86</td>
<td></td>
</tr>
<tr>
<td>flow, 84</td>
<td></td>
</tr>
<tr>
<td>integration, 82</td>
<td></td>
</tr>
<tr>
<td>issue lists, 83</td>
<td></td>
</tr>
<tr>
<td>key users, 63</td>
<td></td>
</tr>
<tr>
<td>performance, 82</td>
<td></td>
</tr>
<tr>
<td>regression, 82</td>
<td></td>
</tr>
<tr>
<td>regression tests, 85</td>
<td></td>
</tr>
<tr>
<td>scenarios, 82</td>
<td></td>
</tr>
<tr>
<td>strategy, 83</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing (Cont.)</th>
<th>success factors, 72 tools, 86 user acceptance, 85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing User Management, 482</td>
<td></td>
</tr>
<tr>
<td>Timeline</td>
<td></td>
</tr>
<tr>
<td>ABAP, 29</td>
<td></td>
</tr>
<tr>
<td>Timing, 41, 207, 218, 330 XML report, 41</td>
<td></td>
</tr>
<tr>
<td>TMS, 112 tp, 230</td>
<td></td>
</tr>
<tr>
<td>training, 64 system, 109</td>
<td></td>
</tr>
<tr>
<td>Transaction logging, 107 Transactions</td>
<td></td>
</tr>
<tr>
<td>existing, 38</td>
<td></td>
</tr>
<tr>
<td>obsolete, 37</td>
<td></td>
</tr>
<tr>
<td>Transport directory, 147, 176</td>
<td></td>
</tr>
<tr>
<td>Transport Management System (TMS), 112, 311 change after copy, 113 change existing, 114</td>
<td></td>
</tr>
<tr>
<td>Transport Organizer, 386</td>
<td></td>
</tr>
<tr>
<td>Transport route for custom objects, 112</td>
<td></td>
</tr>
<tr>
<td>Transports import queue, 316</td>
<td></td>
</tr>
<tr>
<td>TREQ, 455, 472</td>
<td></td>
</tr>
<tr>
<td>SAP Enterprise Portal, 490 upgrade, 553 upgrade documentation, 129</td>
<td></td>
</tr>
<tr>
<td>Trial upgrade, 42</td>
<td></td>
</tr>
<tr>
<td>Trouble ticket, 321</td>
<td></td>
</tr>
<tr>
<td>Troubleshooting, 86</td>
<td></td>
</tr>
<tr>
<td>ABAP upgrade, 320</td>
<td></td>
</tr>
<tr>
<td>Tuning, 218</td>
<td></td>
</tr>
<tr>
<td>Type F error, 303, 312</td>
<td></td>
</tr>
</tbody>
</table>

| Universal Description Discovery and Integration (UDDI), 512 |
| UNIX liveCache upgrade, 436 symbolic link, 150 |
| Unlock system during downtime, 292 |
| Unlock users, 319 unwind, 157 Unzip, 157 Update requests, 264, 286 unprocessed, 265 Update vs. upgrade, 167, 327 Upgrade analyst, 68 approach, 60 business example, 39 coach, 138 complexity of, 43 control document, 133 estimate runtime for production, 41 guides, 126, 549 main phases, 74 media, 154 methodology, 73 minimize the number of, 114 notes, 131 parameters, 227 paths, 535 plan, 71, 74 prerequisites, 76 project, 71 reset, 323 scenarios, 212 schedule, 73 scope, 71 services, 137 strategy, 104, 244 trial, 42, 73 Upgrade Dependency Analyzer (UDA), 45, 138, 420, 456, 474 Upgrade directory, 147, 149, 175, 220, 329 ABAP, 149 change path, 150, 153 htdoc subdirectory, 223 Java, 152 |

| Upgrade directory (Cont.) log subdirectory, 222 tmp subdirectory, 222 Upgrade documentation own document, 133 Upgrade logs, 221 archiving, 223 for error analysis, 324 Java, 338 saving, 300 Upgrade phase run in parallel, 105 Upgrade Planning Tool, 519 Upgrade Portal, 43, 549 Upgrade Roadmap, 43, 57, 88 HTML, 88 Upgrade runtime minimize, 107 Upgrade server access, 229 Upgrade tool synchronization points, 120 UpTime, 211 Used objects, 36 User acceptance, 85 User exists, 35 amount of, 38 don’t work, 116 User ID for upgrade, 229 User Management Engine (UME), 120 |

| Variant Save tool, 117 |
| Variants, 259 problems, 116 restore, 299 with selection values, 259 Verification Check, 406 Version matrix, 535 View, 268 Virtual Machine Container (VMC), 453, 462, 471 activate after upgrade, 480 Virtual system, 112, 113 create as delivery system, 115 |
Virtualization, 418
VirtualProvider, 390
VMC, 453

W
Web Dispatcher, 489
Web Page Composer (WPC), 488
WebClient UI, 453
Where-used list, 316
Wily Introscope, 518
upgrade, 527
Windows
liveCache upgrade, 437
SUM server, 182
Windows Scripting Host, 178
Workplace Plug-in (WP-PI), 391
WPC, 488

X
X Windows, 179
XCM, 466
XI (Exchange Infrastructure), 93
XPRA, 294
errors, 294
postponing, 295

Y
Yes or no factors, 44

Z
ZIP files, 157
Mark Mergaerts is a principal technical consultant working for Logos Consulting, a Belgian consulting company that focuses exclusively on SAP technology (BC). Before that, he worked for SAP Belgium between 1995 and 2011, giving him almost 20 years of SAP BC experience.

Bert Vanstechelman is founder of and principal technical consultant at Logos Consulting, and has more than 20 years of experience in SAP Basis consulting, running all kinds of SAP versions in combination with all possible databases and operating systems supported by SAP.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.

© 2015 by Galileo Press, Inc. This reading sample may be distributed free of charge. In no way must the file be altered, or individual pages be removed. The use for any commercial purpose other than promoting the book is strictly prohibited.