Performing a Successful and Cost-Effective Migration of Your SAP System: Understanding the Export Process

Michael A. Moore



Michael A. Moore manages the Information Technology department for a consumer goods manufacturer in the United States. He has 16 years of experience in both applications development and systems administration. During the past four years, his group has completed two SAP R/3 upgrades, two disaster recovery tests, a disk storage subsystem upgrade, and a successful platform migration.

Has your SAP system outgrown the capability of your current hardware infrastructure? Is your current infrastructure cost too high, or are you under pressure to drastically reduce costs? If the answer to either of these questions is yes, then you are very likely going to be considering an SAP migration.¹

An SAP migration has four main steps: system preparation, data export, data import, and validation. The data export process requires a quiescent SAP system — zero active users, no running jobs, and no scheduled jobs or other activities — as well as a specially tuned operating system and database in order to complete in a timely fashion. The data import process requires configuration of the target operating system, database software, and SAP software, followed by the actual import of the data. Finally, the validation process reschedules all canceled activities and runs new reports to verify the migrated data.

So how do you accomplish all of these tasks — and in the end, a successful migration of your SAP system — quickly and efficiently with minimal risk? One option is to hire a service provider to design a custom solution to fit your needs using native database import/export tools and running export/import processes simultaneously. While a custom solution may offer the shortest system outage time, it can be inherently expensive. If you are migrating less than a terabyte of data, there is a more cost-effective alternative that includes SAP support to minimize risk: the SAP OS/DB Migration Kit.

(complete bio appears on page 24)

For purposes of discussion in this article, an SAP migration will be defined as the export of an SAP database and the import of that data onto a dissimilar target platform. Either the operating system or database (sometimes both) will change during the migration process, but the SAP release will remain the same.

Choosing a Tool for Your SAP System Migration

The first step in your migration project should be to understand the migration process and decide on a migration method. The migration *process* consists of a number of activities: planning, hiring a migration consultant, installing the migration tools, completing an export, moving the data, completing the import, and finally preparing the target system for productive use. The migration *method* is how the actual data exports and imports are accomplished — using either the OS/DB Migration Kit from SAP or a custom-designed solution that uses native database import/export tools. The migration method that you choose will depend mainly on two factors — how much data you have to move and how long the migration outage can last. A typical custom solution is based on native, non-SAP database import and export tools. During such a migration, both the export and import can be completed in parallel across a network connection instead of exporting and importing from files. Since the migration of your data is essentially a one-time process, the most cost-effective solution may be to locate a consultant with experience on your source and target platforms, and let him/her guide you on the choice of migration method.

The OS/DB Migration Kit is an extension of the original SAP R/3 installation tool R3SETUP. It provides extensions built upon R3load to export and then import your data. It also provides a framework that ensures the target system user profiles, file systems, database, and SAP R/3 kernel are installed and configured correctly prior to loading the exported data. Several post-import steps are completed automatically by the migration kit to ensure that you have a responsive and reliable system once the data is migrated. The main drawback to this tool is that it is not designed to allow simultaneous export and import processes, and therefore outage lengths may be suboptimal. The major benefit of using the migration kit is that it is supported by SAP and requires minimal customization to get the job done, which can translate into real cost and time savings. (Be aware, however, that the knowledge of how to use the tool must be provided by an SAP-certified migration consultant; it is *not* provided by SAP support.)

The migration described in this article required that we move 420 GB of data from one operating system platform to another, while using the same DBMS in both the source and target systems. The outage window that we secured for the actual migration was 48 hours (only 34 hours were required). To keep costs down, we chose to use the OS/DB Migration Kit with a single SAP migration consultant familiar with our particular source and target hardware.

This article is the first of a two-part series that shows Basis administrators and IT managers how to use the OS/DB Migration Kit for a successful migration. Using the experience I gained migrating my organization's SAP system, in this article I will introduce the OS/DB Migration Kit. I will explain SAP's migration planning requirements and describe how to prepare the source platform. You will also learn how the export process works and how to improve it to minimize system outage time. In the next installment, I will detail the import and subsequent validation of the migrated system.

Planning the Migration

Before beginning a migration of your system, SAP has three basic requirements that you must satisfy to ensure a stable post-migration environment for your end users. SAP will not support your migration process unless these requirements have been met, and will add charges to the normal maintenance fees to address any problems that could have been avoided by following the requirements. These requirements are:

• A certified SAP migration consultant to assist

with the migration process: Such consultants have been certified to use the OS/DB Migration Kit and to ensure that the customer has SAP support during the migration process. SAP itself does not provide support on how to use the OS/DB Migration Kit — the migration consultant does. (SAP is responsible for providing break-fix repair of the kit, however.) There is a central registry of certified migration consultants located at www.sap.com/solutions/r2/migration/partners. In addition, you may be able to call upon your hardware vendor to provide you with a name (or names) — most large vendors have SAP Competency Centers and will point you to consultants who have been successful working with their platforms.

- A migration plan: The migration plan details all the steps that must be completed prior to the migration of your production system. Your migration consultant must submit the plan to SAP, which in turn will verify that the consultant has the appropriate credentials and is not overcommitted during the project, and that the proposed dates are feasible. SAP will also review the plan to ensure it meets necessary requirements to minimize risk and ensure success.
- The OS/DB Migration Service: SAP provides this fee-based service to qualify your selected SAP migration consultant, evaluate your plan, provide you with the OS/DB Migration Kit, and support you throughout the migration process. You can sign up through your SAP account executive or at http://service.sap.com/servicecat.

Selecting a Migration Consultant

When selecting a migration consultant, it is important to evaluate a consultant's history. You want to choose a consultant who has migration experience on both your source and target platforms, so that you can leverage his or her experience when tuning the export and import processes. Tuning the operating system and database can drastically decrease the overall time required to move the data, as we learned during my organization's migration experience. Obtain past customer references and ensure that you identify the consultant you will be working with by name when you are detailing the services contract. You do not want any last-minute changes to the migration team.

✓ Note!

Although SAP requires that a certified migration consultant be present during the process to minimize risks for the customer, this does not mean that you have to purchase an entire turnkey solution. The migration consultant is required to submit the migration project plan to SAP and to deliver the knowledge of how to use SAP migration tools, but to minimize overall costs you can use your own staff to support the effort instead of hiring additional consultants. For example, during my organization's migration:

- Our SAP Basis administrator was tasked with researching and building the instance profiles for the new SAP landscape.
- Our database administrator was tasked with tuning the database parameters for massive data operations.
- Our operating system engineer was tasked with tuning the export and import environments for efficient disk reads/writes.

Developing a Migration Plan

A previously published *SAP Professional Journal* article takes an in-depth look at how to develop a migration plan.² I will not duplicate the contents of that article here, but the article contains a few important considerations that will affect your overall project timeline and are worth revisiting.

² "A Step-by-Step Guide to Planning for a Successful OS/DB Migration Project" (November/December 2002).

First, SAP requires that both development *and* production systems be migrated to ensure that the developers and testers are working on the migrated system, and resolving any problems, weeks before the end users are. Migrating test systems is optional. If you have plenty of time in your schedule, go ahead and migrate the test system; if you are pressed for time, as we were, you may instead decide to perform a homogeneous copyback (i.e., copy your current production data to your quality assurance and/or test system) once the production system is migrated.³

🖌 Tip

Use the certified migration consultant as a resource to rewrite the copyback procedure for your new infrastructure while he or she is onsite after the production migration. This can save you time and dollars by leveraging their expertise on the target platform.

SAP also requires that the production system be migrated *twice*, and that at least two weeks elapse between the migrations. The first migration allows you to verify consistency between the source and target systems by operating them simultaneously, so that when you perform the second, "real" migration, you can be confident that the daily and weekly batch processing will work as required and as expected.

Finally, once the migration plan is submitted, SAP will require up to two weeks to review it, and then it receives a rating in the form of a green, red, or yellow traffic light icon:

• A green rating, of course, represents a full blessing from SAP.

- A **red** rating means that SAP has found a roadblock in the plan and will not support the migration until the problem is resolved. Having SAP support basically means that if you follow their rules, they will help you resolve post-migration issues without additional charges. Two situations guaranteed to result in a red rating are:
 - The selected migration consultant is not certified on the proper R/3 releases.
 - The R/3 release of the production system was upgraded less than six weeks prior to its scheduled migration. In SAP's estimation, at least one month-end processing period must be completed and evaluated before an upgrade can be considered successful.
- A **yellow** rating means that although SAP has concerns about the plan, they are willing to provide support for the migration. In my organization's case, we received a yellow rating because SAP felt that the schedule was too aggressive. SAP requested that we reconsider the migration dates, but we chose to proceed with our initial plan and I am happy to report that we were successful.

✓ Note!

If you resubmit a plan (because of a red or yellow rating), bear in mind that each such resubmission can add up to two weeks to your total migration project time.

So how long does an SAP migration take? As a general rule of thumb, the data export process will take about one-half of the time required for the overall migration. The primary reason for this is that the source platform is usually an older technology, approaching end of life, while the target platform is current technology. In my organization's migration, we started with a 420 GB database on a three-year-old

³ For more on performing system copies, see the articles "A Homogeneous System Copy in 60 Minutes? It <u>Can</u> Be Done!" (January/February 2002) and "Database System Copies Made Easy — A Guide for Copying an Entire R/3 System to a New Windows Platform" (May/June 2003).

server with twelve 240 MHz processors and 16 GB of memory. The selected target platform was new equipment with eight 668 MHz processors and 16 GB of memory. The underlying storage architecture system did not change — the new server was simply attached to the current Storage Area Network (SAN) with additional space allocated to it. The first (untuned) migration required 26 hours to export the data and another 20 hours for the import. The final production migration took 12 hours to export and 8 hours to import. **Figure 1** summarizes the duration of the migration process and the tasks involved.

✓ Note!

While you may not migrate a production system within six weeks of completing an R/3 release upgrade, as soon as the upgrade project is complete, you can start migration work on the development system, which is upgraded weeks before the production system. For example, due to the nature of my organization's business, it is easier to request and obtain lengthy outages in the spring, so our upgrade and migration projects were scheduled two months apart. As soon as the production upgrade was complete, we submitted the migration plan and started the test migrations. In order to stay within our three-month window, we chose not to migrate our test system (which cut two weeks out of the standard migration plan), and we scheduled the migration of the development system one week prior to the first production system migration. As a result, we were able to maintain SAP support and complete both the upgrade and migration outages during the organization's slow months. Keep in mind that to migrate the development system on time, we had to work on an accelerated schedule of long hours to solve all our migration problems within a fiveweek period. Instead of adding to the complexity of the project and accumulating more risk, you may instead choose to work with your migration consultant and submit a migration plan for "green-light" approval prior to committing to the migration, and save some time in the meantime by completing all the necessary export testing and performance tuning.

Figure 1 Migration Project Timeline

Eva	luating the Migration	Duration
1.	Identify the potential target platform.	20 days
2.	Ensure all required software is supported on the target platform.	
3.	Evaluate potential migration consultants and migration costs.	
4.	Order the OS/DB Migration Kit.	
5.	Install the OS/DB Migration Kit.	
6.	Download all required patches.	
7.	Complete the first export.	
8.	Make the decision to migrate.	
Plar	Duration	
1.	Select a migration consultant.	14 days
2.	Sign up for the OS/DB Migration Service.	
3.	Submit a migration plan to SAP.	
4.	Order new hardware.	
5.	Order an SAP R/3 installation kit for the target platform.	
Pre	paring the Migration Process	Duration
1.	Have the database administrator tune the database for the export.	15 days
2.	Have the operating system engineer tune the operating system for read operations.	
3.	Have the Basis administrator create profiles for the new hardware.	
4.	Have super users and application developers complete the test plan design.	
5.	Receive and install the new hardware.	
6.	Install SAP R/3 on the target hardware.	
7.	Apply patches to the target installation.	
8.	Import the exported data to the target platform.	
9.	Troubleshoot the import process.	
10.	Tune the import process.	
11.	Perform all post-migration system testing.	
Perf	forming the Migration	Duration
1.	Receive SAP approval of the migration plan.	21 days
2.	Perform the final development system migration.	
3.	One week later, perform the first production system migration.	
4.	Test the results, including nightly, weekly, and monthly jobs.	
5.	After a minimum of two weeks, perform the final production system migration.	
6.	Perform a homogeneous system copy to create the target test/QA system.	
Tota	al Project Duration	70 days

🖌 Tip

Prior to committing to a migration, request the OS/DB Migration Kit at http://service.sap.com/ softwarecat. This will allow you to use real export data times when determining if the outage duration will be acceptable to the business prior to committing to the migration. When planning my organization's migration, for example, we completed a test export of the database to be migrated and used that timing to estimate the overall outage length required for both export and import processing.

Signing Up for the OS/DB Migration Service

The OS/DB Migration Service is a fee-based service that provides the framework for ensuring all requirements are met during the migration process. The service includes the following:

- SAP will verify that the selected migration consultant is certified for your SAP release.
- SAP will provide you with a template for creating your migration plan and will review the completed plan for approval (this part of the service is called the *OS/DB Migration Check*).
- SAP will provide an installation kit for your target platform upon request.
- SAP will fully support your migration.

✓ In Summary...

The minimum requirements for a migration using the OS/DB Migration Kit include:

- An appropriately certified migration consultant
- Two weeks for SAP to approve the plan
- A green or yellow rating on the plan
- At least six weeks between an R/3 release upgrade and the migration

Preparing the Source System for the Migration

Once SAP has approved your migration plan, if you haven't done so already, you must complete the following tasks:

- Order an OS/DB Migration Kit⁴ at http://service.sap.com/softwarecat.
- Order the installation kit for the target operating system and database platform (also at **http://service.sap.com/softwarecat**). This kit will be required for the import phase of the migration (to be covered in a future *SAP Professional Journal* article).

✓ Note!

Bear in mind that SAP will ship none of the SAP migration support materials automatically — each must be requested by the customer.

- Check SAP Notes and verify the latest level of database and R/3 kernel software to use on the target platform. Request any necessary patches.
- Check for any additional patch requirements for your target operating system software.
- Generate a migration key at http://service.sap.com/migrationkey.
 Figure 2 is an example of the information required to generate the key, and Figure 3 shows the key returned to the customer based on that information. The target operating system information in Figure 2 is required during the export process; the migration key

⁴ Note that there is one CD for each series of R/3 releases (3.x, 4.0x, 4.5x and 4.6x). For 6.x releases and above, the migration kit is included in the installation kit for the platform.



Information for Generating the Migration Key

Sourcesystem		
System ID	PRD	
R/3 Rel.	Release 4.6C	
OP System	LINUX	
DB System	SAP DB	
DB Server Hostname (case sensitive)	sapprd	
Targetsystem		
System ID	PRD	
OP System	NT/INTELWIN2000	
DB System	INFORMEX	
DB Server Hostname (case sensitive)	sapprd2	
Create Key		

Figure 3

The Generated Migration Key

Migration Key 1W5gh	M50Tv03eatdaQ91aa5	
Expiration date 02-28-	005	
3 Back		

in Figure 3 is required during the import phase to begin importing the data.

- Contact all vendors for any bolt-on software (such as third-party tax software, warehouse interface software, etc.) that you have installed, and request executables for the target platform.
- Determine if any hardware-dependent software keys (e.g., for your database, backup, or other third-party software) will be required during the target system installation.

With these tasks completed, you are ready to prepare your source system to export the data to be migrated. You'll need to:

- 1. Prepare and optimize the source system.
- 2. Install the OS/DB Migration Kit.
- 3. Set up the data export.

We'll look at these steps next.

Step 1: Prepare and Optimize the Source System

Before you install the OS/DB Migration Kit, you need to prepare the source environment by creating space to receive the export files; you may also need to optimize the operating system for the data export.

Reviewing Operating System Parameters

Review the operating system kernel parameters for opportunities to improve disk read performance on the source system. This can have a dramatic effect on export timing. In particular, consider any parameters that control the maximum amount of RAM that the operating system can use for the file buffer cache. Typically, this value is set rather low on a database server to force the database and user processes to use real memory. During the export, the dynamic file system maximum cache size needs to be increased to avoid a potential bottleneck. For example, my organization's source system was normally set to 10%; during the export process we increased that value to 40%.

Work with your operating system support personnel to identify this and other parameters that could affect your export timing.

Two file systems are required for the export process — one for installing the migration kit and another to hold the exported data:

- The OS/DB Migration Kit, which requires 60 MB for its executables and control files, can be located on the same disk as the operating system.
- The export process will extract only data (*not* indexes) and will compress the data before writing it to disk, so a file system with an initial capacity that is approximately 15% of the total space required for the database instance (including indexes, allocated but unused disk space, and data) should be adequate. The export file system must be built locally on or directly attached to the source system (e.g., a SAN or direct-attached DASD). Any attempt to export or import from a remote (e.g., NFS-mounted) file system will result in substantially longer export times and is not recommended.

Finally, it is important that the source operating system and database configurations are optimal for the export process:

• Review the appropriate operating system parame-

ters with regard to file input/output activity (see the sidebar above), such as those that can affect the overall data throughput. For instance, are there any buffers that can be increased, or can you enable the use of more memory for caching disk read activity? The object is to optimize the operating system for increased file input/output activity.

- Review and modify the database configuration to support massive read operations. This may involve, for instance, a review of the database vendor's documentation.
- Review the SAP Notes system for messages related to tuning your particular operating system and database. Since SAP Business Information Warehouse (BW) functionality typically involves massive data reads and writes, you will also want to review SAP Notes pertaining to BW for operating system and database tuning tips.
- Contact your source operating system and database vendors for assistance — their support sites should provide helpful information for making your existing systems run optimally.

Figure 4	Installed Files
Filename	Contents
CEDBMIG.R3S	Central instance/database R3SETUP command file
DBEXPORT.R3S	Database export R3SETUP command file (see Figure 5)
DBMIG.R3S	Database migration R3SETUP command file
INSTGUI	Installation GUI executable
R3SETUP	Executable that interprets and executes the R3S command file
R3ldctl	Load control EXE that creates the data definition language (DDL) template files
R3load	Executable that loads/unloads data from the database
R3szchk	R/3 size-check program to compute table and index sizes
SAPCAR	CAR file bundling/unbundling tool
SPLITSTR.PL	Perl script for splitting structure (STR) files and associated files

Step 2: Install the OS/DB Migration Kit

Once you have received the OS/DB Migration Kit, print out the documentation on the CD. The R/3Heterogeneous System Copy document will guide you through the detailed steps of installing the migration kit and preparing the system for migration (we'll look at the preparation tasks in the next section).

Install the migration kit by following the appropriate instructions for your source operating system. Before running the INSTTOOL command, remember to switch to the directory in which you want to install the migration kit, so that it is the current directory. A typical installation will ask if you want to install the 32-bit or 64-bit version of the code and the type of database to which you will be exporting (Oracle, Informix, etc.).

The installation will then place the files shown in **Figure 4** into the current directory.

These files (with the exception of CEDBMIG.R3S, which will be discussed in the second installment of this two-part article series) and detailed notes on their use will be described shortly.

After installing the files, log on to the SAP Service Marketplace (http://service.sap.com) and download the latest patches for all the files in the migration kit directory. These files will be located in the kernel update section of the R/3 release for your source database and operating system.

✓ Note!

Be sure to apply patches to both the migration kit directory and the SAP program directory /sapmnt/<SID>/exe, because some of the routines will be started by database user sapr3, and the PATH statement for sapr3 may point to the executables in the program directory before the executables in the migration kit directory.

Step 3: Set Up the Data Export

Once the migration tools are installed, it is time to prepare the source system for the export. It is important to understand that the state the source system is in when you begin the export will be the state in which the target system will begin. For instance, if you have a job that has been released and is ready to run when you shut down the SAP system and export the database, that same job may very well start executing

when the target system is started after the data import. Other jobs may be dependent upon an event, such as the initial SAP system start, which would also execute at the start of the target system. Therefore, it is important to ensure that no job is in a released status prior to starting the data export.

Another important aspect of the preparation process is to ensure the integrity of the SAP data dictionary (DDIC). The export process begins by reading the entire contents of the data dictionary and creating sets of control files that contain characteristics about each table in the dictionary. These control files are initially created based on data classes called TABART classes. Every table used by an SAP system is associated with a data class. For every data class, a set of structure (STR), extent (EXT), and *R3load* command (CMD) files will be created that contain detailed information about all the tables defined in that data class.

✓ Note!

Keep in mind that tables that are in the database but not in the data dictionary — such as tables required for bolt-on applications and hence not used directly by R/3 — will not be included in the export. These tables must be moved manually either before or after the export. Also, if the database administrator has moved tables between database spaces without updating the associated data class in the data dictionary, these tables will return to their original database space when they are reloaded. In my organization's case, the psapbtab database space grew tremendously after the migration to the target system. This was directly related to the fact that when the SAP system is initially installed, the largest tables by default are located in database space psapbtab. As tables grew and were relocated to other database spaces, the relationships between the tables and their corresponding data classes were not maintained, and so the tables returned to psapbtab on the newly installed target system.

It is also important to verify that the state of the data dictionary parallels that of the underlying database. Execute transaction *DB02* and run the installation checks for overall data dictionary consistency, missing indexes, or disabled unique indexes inconsistencies may have arisen over time as your R/3 instance was customized. These inconsistencies must be corrected before continuing, otherwise errors may be introduced into the target system.

Lastly, review the notes and the documentation included with the migration kit for specific checks that may need to be carried out on your particular installation. Several checks exist for systems that were installed prior to a particular R/3 release. Others are related to installed modules or certain database or operating system brands. Be sure to perform at least the following key checks:

- Resolve any asynchronous updates that have an error condition (*SM13*).
- If you have implemented operation mode switches, disable them (*RZ04*). Again, the goal is to ensure that nothing will automatically change in the target system when it is first started following the migration.
- If any time has passed since your last SAP Note search, perform one final search in application area *BC-INS-MIGR3*. Perform a wide-open search in this application area and scan the results. This will give you a good idea of the types of problems that you may encounter during the migration.
- Finally, print all SAP module reports that are recommended by the migration kit documentation and any additional reports that your development managers or super users feel will assist them during their post-migration evaluations.

After all of these checks are complete, shut down the source SAP system, but be sure to leave the database running.

✓ Note!

The R/3 Heterogeneous System Copy document included with the migration kit identifies several regularly updated SAP Notes that you will want to review prior to beginning your export. The most important note for an R/3 4.6 migration is 316353 (SAP Basis Heterogeneous System Copy 4.6D). This note will point out fixes to the process, executables, or command files that are required for successful migrations on all operating system and database combinations.

Exporting the Data

Technically, the SAP system is now prepared for the export to begin. Before starting the export, however, you need to understand how the migration kit works.

R3SETUP is the executable that will be running the show. This program works as an interpreter and is called with one parameter — a command file that tells *R3SETUP* what to do. For the export, the command file is *DBEXPORT.R3S*. To initiate the export, the root user issues the following command:

R3SETUP -f DBEXPORT.R3S

The command file is broken into sections that are delimited with headers in square brackets, and each section is followed with a set of detailed instructions that *R3SETUP* uses to determine the actions to execute. **Figure 5** shows a portion of a typical command file.

The most important section to understand is the *[EXE]* section, which controls the overall flow of the export. To improve overall performance, you may have to modify the *[EXE]* section by inserting breakpoints before or after a command is executed in order to stop and restart the export process (see the sidebar "Starting and Stopping the Export Process" on the next page).

Figure 5 Excerpt from a Typical Data Export Command File

```
SAPSYSTEMNAME=@SAPSYSTEMNAME@
SQLSTATS=2
TERMCAP=@DB HOME@/etc/termcap
```

[EXE]

10=DBEXPORTINSTANCE IND IND 20=DBCOMMONPARAMETERS IND INF 30=XIUSERINFO IND INF 40=DBEXPASSISTENT IND IND 50=CHKLIBPATH IND IND 60=DBEXPTESTCONNECT IND INF 70=DBEXPSERVER IND IND 80=DBEXPDIRECTORIES IND INF 90=DBCOMPUTESTAT4MIG IND INF 100=R3LDCTL IND INF 110=R3SZCHK IND INF 120=SPLITSTRFILES IND IND 130=DBEXPCOPYR3LDCTLFILES IND IND 140=DBEXPCOPYEXTFILES IND IND 150=DBEXPR3LOADCMD IND IND 160=DBR3LOADEXECDUMMY IND IND 170=DBEXPR3LOADEXEC IND INF 180=DBGETDATABASESIZE IND INF 190=QUERIESFINISHED IND IND

```
[QUERIESFINISHED_IND_IND]
ACCEPT_KEYS=
```

•••

Once program execution has begun, the *DBEXPORT.R3S* file itself is updated with user responses and success or failure codes for each step. This is helpful from a tracking standpoint, since breakpoints can be inserted and user responses made during those breakpoints can be saved for future export trials and the final migration. By inserting a breakpoint just before the optimization of the export process control files, for example, you can try several variations of the optimization process without reentering all the system data. To accomplish this, insert a breakpoint in the *DBEXPORT.R3S* file prior to the *SPLITSTR* section (shown in bold in Figure 5) or any

Starting and Stopping the Export Process

To optimize export and import processes, you must be able to manipulate the R3S control files to stop and start the export and import processes as needed. The example here has been coded to stop the R3S control file execution just before the SPLITSTR.PL program runs.

Begin by editing the DBEXPORT.R3S file and locating the [EXE] header, which is the control section of the file that will execute all the other sections in the correct order. Under the [EXE] header, locate the string 120=SPLITSTRFILES_IND_IND and insert *above* it the command 115=EXE_STOP:

[EXE] ... 100=R3LDCTL_IND_INF 110=R3SZCHK_IND_INF **115=EXE_STOP** 120=SPLITSTRFILES_IND_IND 130=DBEXPCOPYR3LDCTLFILES_IND_IND ...

Immediately following the [EXE] section, create a new section called [EXE_STOP] with the instruction CExitStep, which tells the R3SETUP interpreter to stop all command processing:

[EXE_STOP] CLASS=CExitStep

other step that you feel could be optimized, as described in the sidebar above. After the *R3SETUP* execution ends, make a copy of the R3S file. If the subsequent steps do not perform to your expectations, stop the execution of the export, restore the saved file, and restart the *R3SETUP* program.

●^{*} Caution!

Some of the export steps cannot be restarted without additional cleanup prior to the rerun. Only by trial and error will you be able to tell where the process can be safely interrupted and restarted.

For each execution of the DBEXPORT process, the existing DBEXPORT.R3S command file is renamed to DBEXPORT.R3S.nn, where nn is a sequential number automatically assigned to each run. In addition to the updated command file, a log file named DBEXPORT.log is created for each run of the program. This log file provides in-depth details about successes and failures as each section of the export is executed. If there is an already-existing file named DBEXPORT.log, the existing file will be automatically renamed to DBEXPORT.log.nn, where nn is a sequential number assigned each time the DBEXPORT process is started. Whenever an error is encountered, the cause will be written to the log file. If processing stops and no errors are logged in the migration kit log file, check the system and database log files.

Next, begin execution of the R3SETUP program using the following command:

R3SETUP -f DBEXPORT.R3S

When the R3SETUP program reaches this step, a message will ask you to confirm that you wish to exit:

Please enter the parameter or confirm default [EXIT]:

Press Enter and the R3SETUP program will exit and return to the operating system.

You can also bypass a step without running it and without stopping the export process, by simply inserting the line STATUS=OK in the appropriate section. In the example below, the section of the DBEXPORT.R3S file that executes the update statistics process is skipped:

```
[DBCOMPUTESTAT4MIG_IND_INF]
CLASS=CExecUpd
...
RUN_AS_USER=@INFUSER@
STATUS=OK
STEP ENV=DB ENV
```

This technique can be used to shorten the overall system outage time required for the migration (more on this later in the article).

These steps are explained in greater detail in SAP Note 118059.

🖌 Tip

The most recent log and command files can be located by sorting the migration kit directory by the last time the file was updated.

Now that you have a solid understanding of how the migration kit works, let's walk step by step through what happens once the export process is executed.

Step 1: Execute the Export Command File

Before you actually start the export, you will need

to set up the database environment variable *LIB_PATH* in the current shell. This configuration is dependent upon the source database and operating system and is detailed in the *R/3 Heterogeneous System Copy* document included on the migration kit CD.

If you are going to monitor the export process using the *INSTGUI* tool,⁵ start it now as described in the section "The R3SETUP Tool" in the *R/3 Installation on UNIX:Informix* document included on the migration kit CD. The root user environment and migration tool are both ready, and it is now time to execute the export command file.

⁵ INSTGUI is a graphical tool that allows you to view the progress of the migration and any errors that are generated.

Fiaure	6
igaio	~

Specifying the SAP Instance Information

```
1 Enter/confirm the SAP System ID (three-character string in
  uppercase). UPG
 2 Enter/confirm the SAP Instance Number (two digits). 00
 3 Enter/confirm the SAPMOUNT directory: /sapmnt
 4 Enter/confirm the name of the SAP central instance host. userupg
 5 Enter/confirm the name of the SAP database server host. userupg
 6 Enter 32 to install the 32 bit version of Informix Backend, or
   enter 64 to install the 64 bit version. 64
 7 Enter your target operating system. 3
   (1) ALPHA
   (2) AS400
   (3) AIX
   (4) DEC
   (5) HP10
   (6) HP11
   (7) I386
   (8) LINUX
   (9) PTX
   (10) RLX
   (11) SUNOS
 8 Enter your target database: 5
   (1) SAP DB
   (2) DB2/390
   (3) DB2/400
   (4) DB2 UDB for UNIX, Windows
   (5) Informix
   (6) Microsoft SOL Server
   (7) Oracle
 9 Enter the export directory $EXPORT DIR. The installation tool
  creates the CD-Tree $EXPORT DIR/DB and $EXPORT DIR/DATA and
  writes the file LABEL.ASC. /dbexport
10 (1) Continue the export. ATTENTION: Migration key will be needed
   for import! 1
11 (1) Yes, split STR and EXT files. 1
               put the N largest tables in separate files 1
12 (1) -top:
13 number of tables to be put in separate STR files: 5 (default)
14 Enter/confirm the maximum size of an export dump file. Attention:
   This step does not overwrite existing .CMD files. So make sure
   that the installation directory does not contain any old .CMD or
   .LOG files before performing an export. 500
15 Enter/confirm the number of parallel processes for the database
  load. (2.5 times number of processors)
16 Enter/confirm the maximum size of an export dump file. 500
```

Step 2: Enter the SAP Instance Information

enter the SAP instance information. The question and answer session shown in **Figure 6** (the answers are shown in bold) are from an export

The export process will first request that the user

of my organization's test system.⁶ The system identification (SID) was UPG, the SAP system number was 00, and the hostname of the system was *userupg*. Both the database and R/3 central instance resided on the same server.

Notice that the target system information is requested, but information regarding the platform you're performing the export on is not. The migration kit will automatically determine the source platform information, and then (based on the source and target characteristics entered during the Q&A session) generate the migration key that is stored with the export files. The migration key that was generated at **http://service.sap.com/migrationkey** (refer back to Figures 2 and 3) must match the one stored with the export files in order for the import process to begin on the target system.

✓ Note!

Be sure to enter the target information precisely. If there are problems importing the exported files on the target system, do not assume the files are bad — it instead may be the result of an invalid migration key. The migration key was generated incorrectly for my organization's combination of source and target platforms, for example. Before spending hours performing additional exports, contact SAP to verify that you have correctly specified your platforms and there is no problem with the migration key generation.⁷

At this point, the export proper begins. The export process first checks the entered export parameters for validity, including checking root user environment variables (*LIB_PATH*), verifying the database installation, and making a test database connection using *R3trans*. Export preprocessing finishes by creating the export subdirectory structure that will be used by the export processes.

Step 3: The Database Table Statistics Are Updated

Since databases store records in chronological order as they are added to the table, the most efficient way to export the data is to dump it in the order in which it was stored in the table, which would be termed an *unordered* export of the database. However, the migration kit exports all tables in an *ordered* fashion, using an index and associated primary key value, by asking the database instance to run an update statistics process against every table so that all statistics will be rebuilt. By creating current statistics, the migration kit can be assured that the most efficient SQL plans will be used when exporting the data.

The migration kit updates statistics for all tables in the database by issuing the *sapdba* command:

/sapmnt/UPG/exe/sapdba -updstat
 -forcedlow

✓ Note!

The statistics generation process may add several hours to a typical database export. Please refer to the "Helpful Hints for Improving Export Timing" sidebar on pages 22-23 for methods that may help you to avoid this additional time.

Step 4: The Structure Files Are Created

Once the statistics are updated, the *R3SETUP* process creates the *SAP*<*TABART*>.*STR* files (database-independent structure files) that are used to generate

⁶ Line numbers have been added for ease of reference and to aid in readability — they do not appear in actual practice.

In reality, this particular error rarely happens. I mention it here so that if you are testing the migration process and a small error occurs, you know that you should take it to SAP support immediately instead of trying to resolve it yourself. SAP solved my organization's problem in five minutes, after we had spent two days performing unnecessary additional exports.

Figure	7
riguic	/

Excerpt from a Typical Structure File

tab:	A055							
att:	APPL1 1	T all	A055~0				APPL1 1	
fld:	MANDT			CLNT	3	0	0 not_null	1
fld:	KAPPL			CHAR	2	0	0 not_null	2
fld:	KSCHL			CHAR	4	0	0 not_null	3
fld:	VKORGAU			CHAR	4	0	0 not_null	4
fld:	WERKS			CHAR	4	0	0 not_null	5
fld:	MATNR			CHAR	18	0	0 not_null	6
fld:	DATBI			DATS	8	0	0 not_null	7
fld:	DATAB			DATS	8	0	0 not_null	0
fld:	KNUMH			CHAR	10	0	0 not_null	0

unload scripts. The *R3ldctl* (load control) program is executed with a single parameter that indicates the path in which the structure files will be created. The program then logs in to the database as database user *sapr3* (remember that the operating system *sapr3* user must have the same password as the database *sapr3* user), inspects all data class definitions, and for each data class found, writes table definitions to the corresponding structure files.

All pertinent information is documented in the corresponding structure files for every table to be exported. **Figure 7** is an excerpt of an actual structure file, taken from my organization's test system:

- The *tab*: line indicates the start of each table definition section in Figure 7, the table is *A055*.
- The *att:* line documents several table attributes, as shown in Figure 7:
 - The first *APPL1* is the data class (TABART).
 - The first *1* is the table category (TABKAT).
 - *T* (codepage) and *all* (flag) specify the conversion type.
 - $A055 \sim 0$ is the name of the primary key.
 - The second *APPL1* is the index data class (TABART).

- The second *1* is the index table category (TABKAT).
- Secondary indexes, if they exist, are defined using the keyword *ind*: (there are none in Figure 7).
- Fields and their attributes are documented using the keyword *fld:* followed by the field's attributes (name, type, length, etc.).

Review the *tab:* and *att:* lines to determine the table name and associated data class (TABART) as defined in the SAP data dictionary. If a database space on the target system following the import process is much larger than the corresponding database space on the source system, the reason can readily be determined by analyzing the content of these files.⁸

Step 5: The Template Files Are Created

Next, the *R3SETUP* process creates the *DDL*<*DB_TYPE*>.*TPL* template files, which contain the data definition language (DDL) that will be used to export the tables. Each brand of database software (Informix, Oracle, SQL Server, etc.) requires a different DDL for completing ordered exports.

⁸ See SAP Note 46272 (Implementing new data classes in technical settings) for additional information.

A template file contains several sections:

- The data definition template sections are used to create the tables, key rules, and indexes.
- The *loc:* (location) section details the data class (TABART) and its corresponding storage group. For instance, all tables of class *APPL1* will reside in storage group *BTAB*, which corresponds to tablespace *psapbtab* (regardless of the DBMS being used).
- The *sto:* (storage) section details the characteristics of each data class and the corresponding extent characteristics, which vary depending on the database vendor being used.

The template files should not be modified except under the guidance of SAP support.

Step 6: The Extent Files Are Created

The next program that runs is *R3szchk*, which reads the contents of the structure files and creates the corresponding *SAP*<*TABART*>.*EXT* (extent) files in the current directory. These files contain the name and initial size of each table and its associated indexes. The following code sample shows entries in the *SAPSSRC.EXT* file for table *CROSS*, which has secondary indexes *CROSS~0* and *CROSS~I*:

CROSS	323631513
CROSS~0	161814118
CROSS~I	28889907

These extent files are created for all structure files except for one, *SAPVIEW.STR*, which is used to create all the views that are present in the SAP R/3 system. Remember that views do not have corresponding data but are joins of selected tables.

Step 7: The Command Files Are Created

R3ldctl now runs a second time, using the structure and extent files to create *SAP*<*TABART*>.*CMD* (command) files, which are used to control the export

Figure 8 An Excerpt from a Command File

icf:	/dbexport/DATA/SAPSSRC.STR
dcf:	/dbexport/DATA/DDLINF.TPL
dat:	/dbexport/DATA/ bs=1K fs=500M
dir:	/dbexport/DATA/SAPSSRC.TOC
ext:	/dbexport/DB/INF/SAPSSRC.EXT

process for each group of tables. **Figure 8** shows a portion of a command file.

Command files are formatted with labels and directives:

- The *icf*: (independent control file) label indicates the path and name of the structure file.
- The *dcf*: (database-dependent control file) label indicates the path and name of the database template file that contains the data definition language used to export the data.
- The *dat:* (data dump) section details the characteristics of the data files, including block size and maximum file size.
- The *dir:* (or directory) section points to the table of contents (TOC) file. The TOC file is created during the export and contains a listing of all corresponding data dump files for all tables listed in the structure file.
- The *ext:* (or extent) file points to the corresponding initial extent file.

At this point, all command files required for the export process have been created.

✓ Note!

SAP Note 118059 contains additional information on modifying structure, extent, command, and template files.

Step 8: Split the Default Structure and Extent Files

Next, you have the option of splitting the default structure and extent files, which can help to spread the export load over time as efficiently as possible.

Each *R3load* process that runs will export data for all tables in a particular structure file. In my organization's system, the largest tables were physically spread over several database spaces, but the SAP data dictionary did not understand this because the tables were still associated with their default data classes. (By definition, each table has a one-to-one relationship with a data class.) This resulted in the largest tables being included in the same structure file and the table exports being executed sequentially instead of in parallel.⁹ To improve the overall performance, we used a *SPLITSTR.PL* script to split the tables into new *R3load* file sets based on the table size.

You have two options for executing SPLITSTR.PL. The R3SETUP program can execute SPLITSTR.PL for you and place the largest tables into separate files, or you can run SPLITSTR.PL manually, which allows you to separate the files in different ways: maximum MB of data to export per process, or maximum number of tables to export per process.¹⁰ The most efficient method for us was to stop the export process at this step and manually run the SPLITSTR.PL program, limiting each set of R3load command files to processing a maximum of 500 MB of data each. Tables larger than 500 MB were placed in command files by themselves. If you are still working on your first successful export, I recommend that you let the R3SETUP program execute SPLITSTR.PL on your behalf and place the top five files into separate structure files. The first successful export will provide you with a base time against which you will be able to gauge the effectiveness of your export tuning efforts.

Step 9: Start the Export

Once the *SPLITSTR.PL* program has completed, the export process will begin. If you halted the export process, restart it by changing back to your migration kit installation directory and re-executing the command:

R3SETUP -f DBEXPORT.R3S

Based upon the number of processes you have allowed to run in parallel, several *R3load* processes will be started. For example:

R3load -e SAP<TABART>.STR -p SAP<TABART>.log

As each *R3load* process ends, another starts, until all structure files have been processed. If for any reason an *R3load* process fails, the export can be restarted — just let the running processes continue, and when the remaining processes have finished, restart the export.

The *R3load* process will begin to export the tables that are listed in the structure file in sequential order and will create a log file in the migration kit installation directory. The data files that each *R3load* process creates are named SAP<TABART>.nnn, where nnn is a sequential number assigned to each new file created because the previous file exceeded the maximum size. Each data file is created in a directory called DATA located in the specified export directory (/dbexport in Figure 6). In addition, each data file will contain only the amount of data that was specified (500 MB in Figure 6). For each structure file, a table of contents file (SAP<TABART>.TOC) will contain a listing of tables and associated data files, as well as the location of the exported data within the file. Figure 9 contains a sample of such a file.

Figure 9 An Excerpt from a Table of Contents File

1

⁹ Lack of parallelism can have a huge negative impact on migration throughput. Early on, while we were still learning the migration process, we started a data export that was still running over two weeks later!

¹⁰ See SAP Note 200044 (Splitting STR files) for additional information.

Each TOC file contains information about the file (or files) to which the data has been exported:

- The *tab:* row indicates the table name.
- The *fil:* row first indicates the name of the file to which the table data has been exported. This same row also indicates the block size of the file (1024 in Figure 9) and ends with a comment indicating the date and time the export was started, using the format *#YYYYMMDDHHMMSS*.
- The next row indicates the beginning and ending block numbers in the file and the number of rows that were exported.
- The *eot:* row indicates the end of table information and includes another comment with the date and timestamp for when that particular table export completed.

After all *R3load* processes have been executed, the final file that will be created is the *DBSIZE.TPL* file in the /<*export_dir>/DB*/<*db_type>* directory. This file contains characteristics about the containers in which the tables will be reloaded on the target system. **Figure 10** shows a portion of a *DBSIZE.TPL* file created during an Informix database export. In the example shown in Figure 10, the database space name *psapsystem* (shown in bold) will be created with 300 MB of data space, and the database instance will create the database space mirror by allocating 600 MB (2×300 MB) of raw disk space and mirroring the data using database utilities.

✓ Note!

The DBSIZE.TPL file may need to be changed if you are using a disk storage system that presents protected disk (mirrored or RAID 5) to the operating system. In this case, manually edit the file and change all occurrences of the keyword MIRROR to NOMIRROR.

Congratulations — if everything worked as planned, your first export should now have run to completion! While the overall export will probably last much longer than you desire, based on my experience, there are a few easy steps you can take to dramatically improve your timing. The sidebar on the next page outlines these steps.

Figure 10

An Excerpt from DBSIZE.TPL

Helpful Hints for Improving Export Timing

✓ The first timesaver is simply to run an update statistics process prior to beginning the migration outage (we did this the weekend before the outage). Ensure that this pre-outage update statistics run will update the statistics for *all* tables in the database. Then, as the migration outage begins, force an update statistics run on all tables that have changed more than 10%. The first update statistics run may take a very long time and may affect end-user performance. However, the second update statistics run will execute quickly and only generate new statistics for those tables with major changes.

To do this, have your database administrator perform an update statistics run using the SAPDBA tool, specifying the option *-forcedlow* within a week of beginning the migration. Then, before starting the export process, modify the *DBEXPORT.R3S* file by locating the *[DBCOMPUTESTAT4MIG_IND_INF]* section and changing the line:

OPTIONS= -updstat -forcedlow

to:

OPTIONS= -updstat -threshold 10 -level high.

This command will search for all tables that have changed more than 10% (*-threshold 10*) since the last update statistics run and perform a high-level update statistics run (*-level high*) on both the heading and non-heading index columns. (The main difference between *-level high* and *-level medium* is the number of rows sampled in the non-heading index columns for calculating the statistics that will be used to optimize the export SQL plans.) The combination of the two update statistics processes will result in efficient exports with less time required during the outage.

✓ Split the structure files in more complex ways than simply placing a number of the largest tables into separate structure files. Stop the export process (see the sidebar "Starting and Stopping the Export Process" on pages 14-15) just after all the files have been created and before the export processes begin. Now use the SPLITSTR.PL command to manually split the structure files into groups that contain no more than 500 MB of data each. I recommend using the command:

SPLITSTR.PL -p <target path> -d <source path> -limit 500

Conclusion

The export process presented here can be used to estimate the required time for the migration outage. After completing the first export, apply the necessary performance enhancements to minimize the amount of time that is required to export the entire database. Don't forget to monitor and tune the operating system and database. Talk with your migration consultant to discuss any additional steps that may improve your data extract process. The experience he or she has gained by participating in several previous migration projects should serve to decrease the time for your export. Perform a couple of additional exports in order to determine the most efficient number of processes used to export the data. Typically, if you are using at least 95% of the CPU resources of the system, but are not context switching or paging, then you are about optimal. You don't want to create a situation where you are slowing the export process In our case, this resulted in creating structure files that contained sets of tables with a sum total size of no greater than 500 MB. Therefore, if an individual table was larger than 500 MB, it was forced into a structure file by itself. Notice that this command requires a target and source path to be defined. The *SPLITSTR.PL* program will not overwrite the original structure and command files, but instead creates new files that will be located in the specified target directory. After splitting the structure and command files, you must remove the original structure and command files from their default location and move the newly defined files into the export directory so that the export process will use the newly created files.

- ✓ Be aware that the order in which the tables are exported is very important. Imagine if the largest table export were started as the last export process the export would take much longer than necessary. It is best to have the largest tables begin exporting at the very beginning of the actual export process. While the export process has been halted for splitting the structure and extent files into more reasonable groupings, take advantage of this time to force the largest table exports to start at the beginning of the export process. In the Unix operating system, *R3SETUP* processes the structure files in the actual order that they have been created in the directory. After much experimentation, we found that the easiest way to force the execution of the files with the largest tables started with a simple rename and copy process:
 - After splitting the STR and EXT files into appropriate groups, go to the directory containing the updated STR and EXT files and identify the files that contain a single table. These files will have the largest amount of data to export, and export of their data should be started near the beginning of the export process.
 - 2. Rename these files so that they are guaranteed to be listed first (*SAPA*<*TABART*>.*STR* instead of *SAP*<*TABART*>.*STR*, for example).
 - 3. Copy the STR and EXT files back to the migration kit installation directory sorted by name, which will force the execution of the largest files first.

In theory, if the tables are grouped appropriately and the largest tables are isolated into individual export processes scheduled to start immediately, then your complete export time should equal the time required to export the largest table. Of course, if you are running on an undersized system, you may not be able to achieve this result — nevertheless it is the goal to shoot for.

down because you are attempting to overuse the system resources, but you do want to push the system to the edge.

In the end, if your database export has been effectively tuned, the total outage time should be just over twice as long as the export time. The reason for this is that the target system will typically be at least two to three years newer than the source system. As such, the speeds and feeds of the data import should be much faster than the export. However, additional time will be required to physically move the data files across the network from the source to the target system, perform the index rebuilds after the import, complete the first update statistics run on the target platform, regenerate all program loads for the target environment, and perform post-migration system checks. These steps will be covered in detail in the second installment of this two-part article series. Before signing up for the OS D/B Migration Service and purchasing hardware, I recommend that you invest the time and money in completing an export. For my organization's migration project, I was responsible for ensuring that all risks were identified and that the migration occurred in a timely fashion. How could one expect any manager to do this without a thorough understanding of the process? My hope is that by outlining the effort and risks involved, this article has provided the direction you need to ensure a successful migration.

Michael A. Moore manages the Information Technology department for a consumer goods manufacturer in the United States. He has 16 years of experience in both applications development and systems administration. Mike started working with SAP in January 2000, and during the past four years, his group has completed two SAP R/3 upgrades, two disaster recovery tests, a disk storage subsystem upgrade, and a successful platform migration.

Prior to his SAP experience, Mike was a senior operating system engineer for a national insurance company, where he was responsible for Unix systems located all over the East Coast of the United States. While there, Mike specialized in installing and managing Unix servers and applications requiring high availability, including data warehousing and imaging. Mike also has eight years of application development and software quality assurance experience gained while employed at various federal nuclear installations. Mike is a co-holder of a software patent (along with his former employer) for an application design that manages complex storage arrays. He can be reached at MikeAMoore@aol.com.