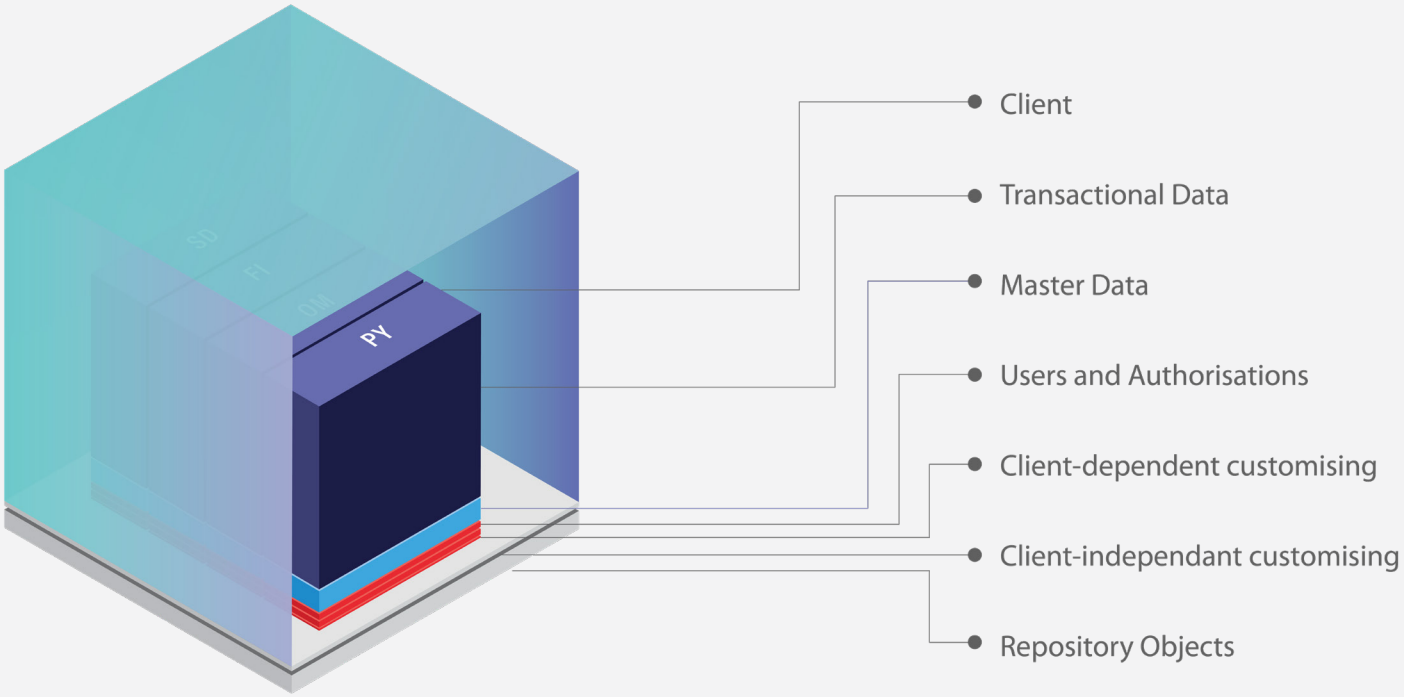
LEVERAGING SYSTEM BUILDER BEFORE, DURING AND AFTER AN S/4HANA MIGRATION

Paul Hammersley

One of the reasons SAP became the leading ERP solution between 1990 and 2010 was the ability to configure and tailor it to a wide range of processes, encompassing almost all industries. A big benefit of the system is that the different business areas are tightly coupled. If stock left the warehouse, the finance data was updated automatically, for example.

System Copy: what is it, and why is it needed?

Traditional ABAP stack SAP systems have, at a high level, five distinct layers.



1. At the base there is the Repository. This is the table definitions, program code, etc. which provide the ABAP runtime environment. Some of this data is a mirror of what the underlying database stores, but typically with an extra layer of information such as foreign key relationships.
2. Client Independent Customising is a small layer of configuration which is shared by all clients of the SAP system. Common examples are Printer definitions and RFC destinations.
3. Client Dependent Customising, also known as configuration, are the settings which define how business processes work, how enterprise structure elements interact, etc. In other words, everything which determines how the data will function.
4. The Application data is made up of Master Data which is relatively static and is typically valid for a wide period of time, and Transactional Data: the day-to-day activities of the business which typically reference one or more pieces of master data.
5. User master and authorisations data controls the data users can see and create in SAP.

One of the reasons SAP became the leading ERP solution between 1990 and 2010 was the ability to configure and tailor it to a wide range of processes, encompassing almost all industries. A big benefit of the system is that the different business areas are tightly coupled. If stock left the warehouse, the finance data was updated automatically, for example. Due to the critical nature of SAP for the vast majority of its customers' businesses, changes cannot be made lightly. So layers 1) and 3) above can only be changed in a development system, and are then 'transported' to testing and eventually Production if they are successful.

The challenge for customers is then how to provision accurate testing environments to ensure that as 1) and 3) change, particularly the fourth layer (4) is indicative of the current processes and data. Just updating 4) from a Production environment is no simple task, particularly given the volumes. Most organisations running SAP use a periodic full copy of Production onto at least one of their test systems. They then go through a number of post-processing steps to revert 3) and 5) back to how those layers were before the copy, and then reapply 'Work in progress' transports that had previously gone to the test system but not Production, and so have been reverted in the copy.

Heterogeneous System Copy compared to System Builder

In modern times, hardware replication technology is used to restore snapshots of entire servers or virtual machines.
The majority of Basis consultants, internal and external, become familiar with several of these approaches as they work on sites with different operating systems and database types. All of these approaches fall under the term Homogeneous System Copy, because the Operating System and Database type does not change.

However, there is another type of system copy which has been forgotten, or never learnt, by a lot of Basis teams;
that is Heterogeneous System Copies. One of the founding principles of SAP was that customers should have a choice of Operating System and Database, effectively creating a competitive market to ensure value for their customers. When installing a new SAP system, the correct installation files for operating system type and version must be downloaded. Then there are parts of the installation media which are generic, and parts which depend on the database type. This media runs what was previously known as the SAPInst process (now the SAP Software Provisioning Manager (SWPM)): a lightweight JAVA-based application which could run almost identically on any of the permitted operating systems.

SAP transport methodology between ECC clients

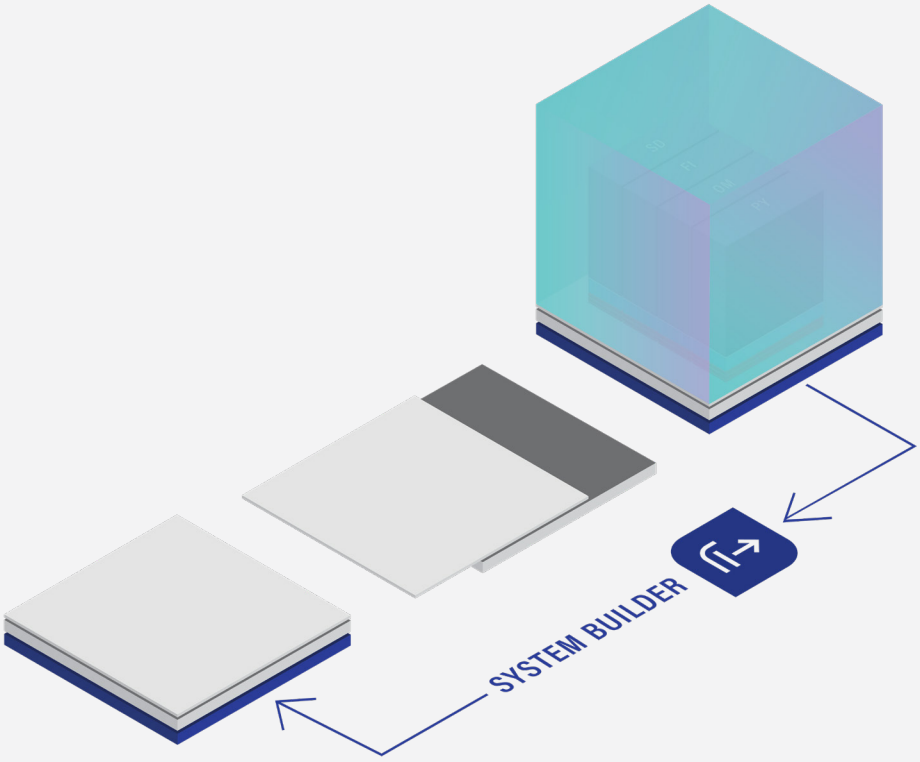
Once the database server and then application server(s) had been installed, a 'vanilla' or 'clean' SAP system was in place and ready for the project to begin. This installation process designed for the initial installation could also be used when companies decided to move from one OS/DB combination to another (the Heterogeneous System Copy). SWPM or SAPInst would run at the operating system level on the source system and leverage the r3load executable which forms part of the SAP kernel to export the entire system into load files that actually correspond to the vanilla loads used for initial SAP system creation.

These load files are operating system and database agnostic, allowing the user to move them to the new host (potentially on a different operating system) and run SWPM to install the system following the options for a system conversion rather than a new installation – the media being used for the installation being specific to the new OS/DB not the old one. At the end of the process, the entire system is now up and running with the same SAP SID on a new platform. The process required a SAP-certified migration consultant to be involved in the project, and a key was required to begin the installation step.

System Builder

Returning to our five layers of an ABAP stack system:

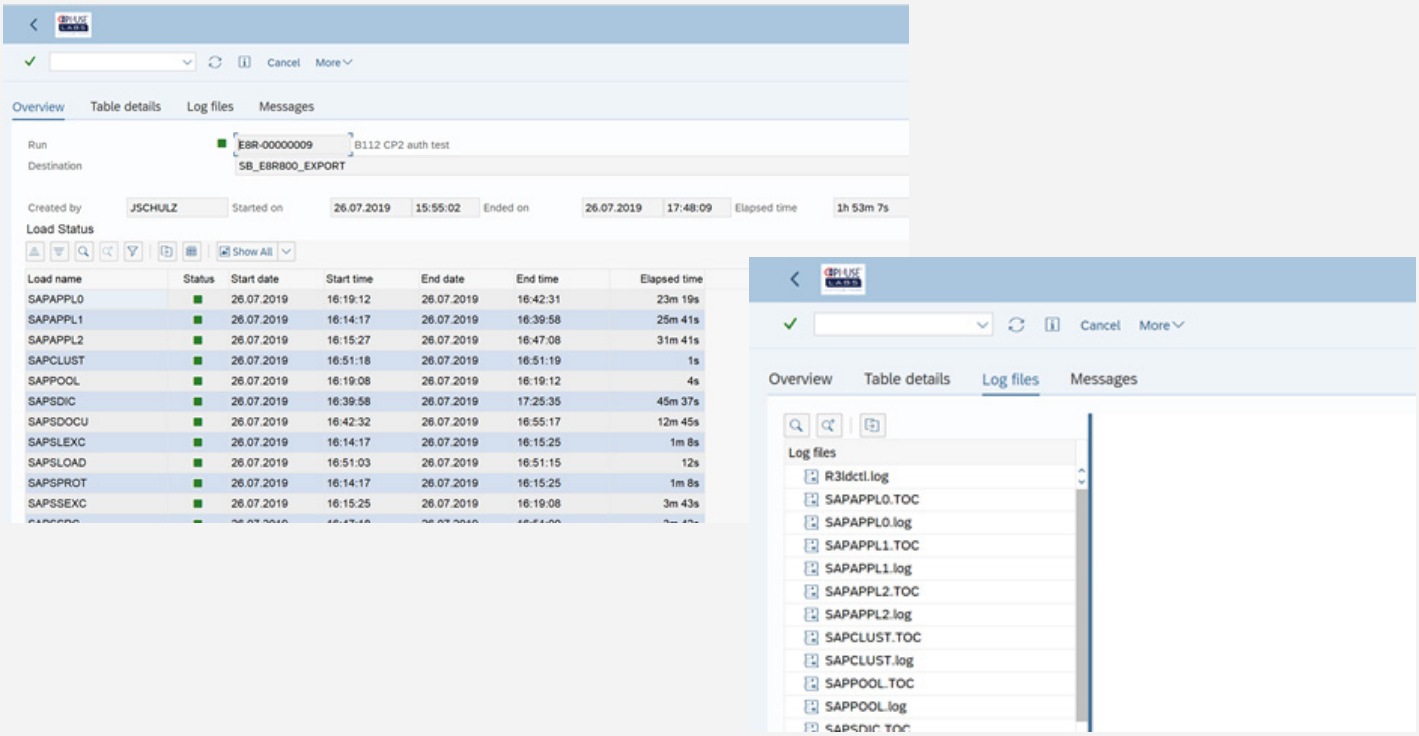
1. Repository
2. Client-independent customising
3. Client-dependent customising
4. Application data
5. Users and authorisations



3, 4 and 5 could be copied from Production to test systems via a client copy, but only extremely small systems could allow this because of the downtime required on the source system. Third-party solutions like Client Sync™, part of the EPI-USE Labs Data Sync Manager suite, made this possible for much larger systems because they excluded unnecessary tables, and reduced the volumes in the biggest tables by only taking data for specific company codes and related structures, and/or limiting the history of transactional data to the last x months. This allowed more efficient refreshes of test systems, and in many organizations replaced the system copy process completely.

However, in certain circumstances the first and even second layer might also be misaligned and could hamper accurate testing. This was more common on OLAP systems like BW and SCM where some of the configuration is client-independent, but even on the OLTP systems there could be specific projects requiring a stand-alone new test system, or if a parallel development track is used for projects the repository must be reset at the start of each new cycle. Copying the whole system, then deleting 3, 4 and 5 to then import the reduced set from Production is not an efficient process due to the need to house the full data set on the new system temporarily, and the time it can take.

System Builder mirrors the Heterogeneous System Copy process in how the new test system is installed using SWPM, but replaces the export phase with a process executed from within the SAP system and managed by SAP authorisations.



An optimisation program calculates the 'splits' required on what will be the largest tables beforehand; then when the process is executed, background jobs at the SAP level trigger r3load processes at the operating system level to extract all the repository and client independent tables including their content. The client-dependent customising, application data and user/authorisations tables have their definitions extracted but none of the content except for clients 000, 066 (if it exists) and optionally client 001. During the installation with SWPM the new SID is set and when that process completes the system has all the custom tables, code and client independent configuration but without the Production client, thus providing the smallest possible shell for a Client Sync or client copy into.

Preparing for S/4HANA: phase zero or the project start

The requirements for carrying out a brownfield migration to S/4HANA can vary quite widely between different organisations, depending on the industry, master data, functionality implemented and customisations made in the system. Some preparation work to understand the level of effort required can be done without beginning the project, but to see the full picture, SAP recommends a Sandbox migration at least three months before the project is due to start, and warns that the conversion may need to be repeated two or three times before it is successful. This is challenging from a business perspective, because the outcome of that sandbox project could fundamentally challenge the initial decision to go brownfield rather than greenfield.

Some experts advise a 'phase zero' of the project where no firm commitment has been made to greenfield or brownfield. That may even include running the sandbox project. The lifespan of this sandbox is very hard to predict, and while it is used, the existing business-as-usual support and other projects will still be taking place. Therefore additional hardware will be needed in most cases, which given the temporary nature makes a perfect use case for cloud, even if the existing hardware is in house. The hardware required to go through the S/4HANA conversion may also be to a higher specification than the existing sandbox/test systems. SAP Tailored Datacenter Integration (TDI) allows a much wider range of servers to be used to run HANA, but they are still typically much more powerful than those traditionally used with ERP systems as a result of the in-memory requirements. An existing SAP ERP system with that much memory would have been much more than was required for a traditional RDBMS.

When planning such a sandbox, it is important to consider what the starting system should be. Conversion issues which only affect a development system, or test system, are not likely to be significant if they do not also affect Production and could be worked around. The closer the sandbox aligns with Production, particularly for layers 1), 3) and 4) the better the quality of the information that will be gleaned from the project. So this returns us to the same challenge we discussed earlier, how to replace layer 1) with data from Production without needing to copy the entire system. Particularly given the impact of system size on the cost of temporary or cloud hardware, it makes no sense to use a full copy of Production.

System Builder and Client Sync can be used in conjunction to provision an accurate representation of the system, but typically around a quarter of the size. This means either a smaller capital outlay for the hardware, or a reduced ongoing cost for cloud consumption. If Client Sync is already leveraged by the organisation, an alternative would be to copy the full system and then delete and recreate the client, but the temporary need for the full volume may be expensive or prohibitive if hardware is to be purchased since it will only need to be that size for a few days. The alternative of copying a development system (since these are smaller) may mean a lot of additional code remediation and false positives in the S/4HANA conversion testing due to an abundance of code that never made it to Production, and data which is not representative at all of the business processes now used in Production.

Building the landscape for S/4HANA conversion

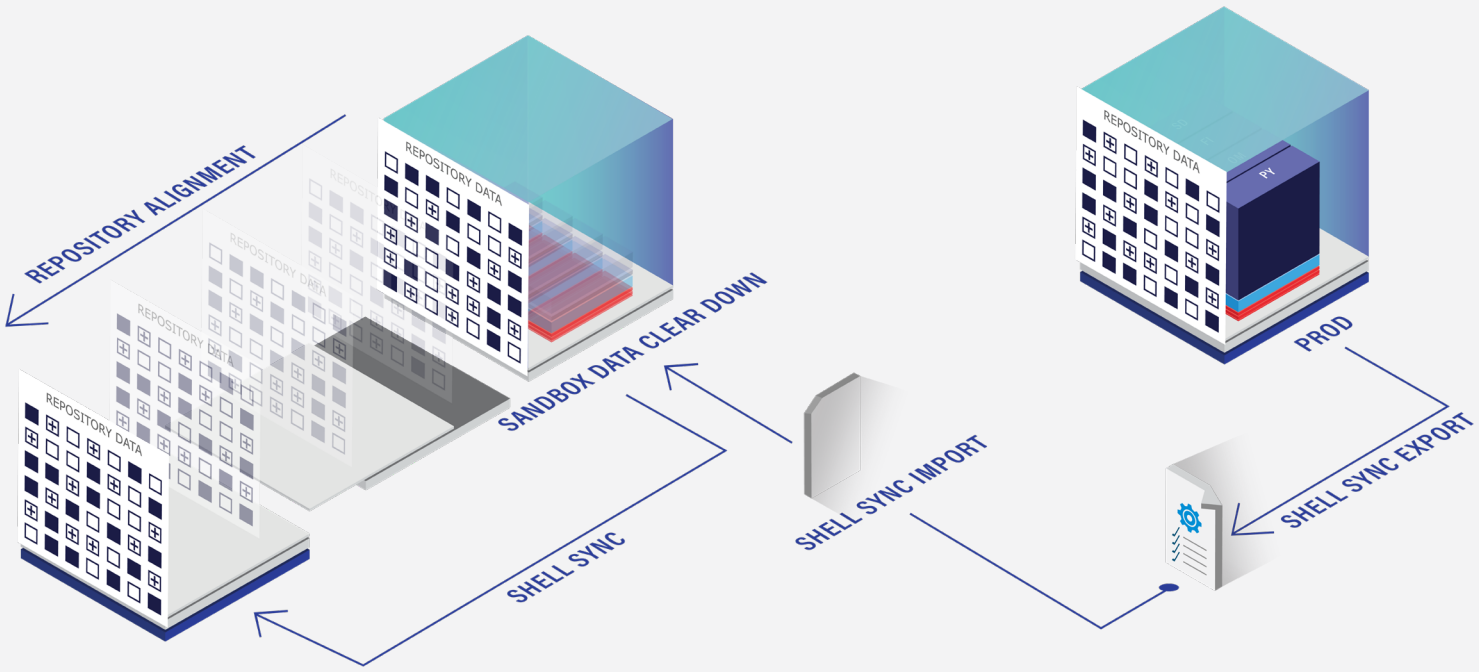
The difference between the repository in Development and Production is also a consideration during the conversion project itself. For the majority of SAP customers, the Development system is more than a decade old and has never been updated as a system copy because of the desire to keep version history on code. This means a significantly larger job carrying out the code remediation steps needed for S/4HANA in Development than will be needed in Production. This has led some organisations to jettison the Development and Test systems and make full copies of Production for both in the landscape that will go through the conversion process.

This means a considerable development freeze in the final stages of the project or dual maintenance of changes for a period of time. The hardware costs for both Test and Development being the same size as Production are heavy and will be hard to reduce once the project has completed without losing the quality of test data required. The combination of System Builder and Client Sync can again be used to mitigate this significant cost. A very small amount of data can be included in the Development system with a larger subset used for the Test system, probably aligned to the size of the Sandbox reduction.

The Version History of ABAP code is important and traditionally the reason why organisations don't replace their Development System with either a full or lean copy from Production. This can be remedied using the steps outlined in SAP Note 130906 to export the relevant tables from the old Development System and load them into the new one, whether that was created as a full copy or via System Builder and Client Sync.

Transitioning to Shell Sync

The new S/4HANA world customers are steadily moving to does not include the same freedom to choose between different platforms that was previously there. While application servers can also be Windows based, the database server is Linux, and the database of course HANA Database (HDB). As a result the Heterogeneous System Copy we discussed earlier is no longer required, and from the kernel shipped with S/4HANA 1709 no longer supported. As a result, the System Builder product is not available for S/4HANA 1709 version and above.



The replacement for System Builder, Shell Sync™, addresses the challenge in a slightly different way. Uninstalling and reinstalling SAP was quite onerous for customers that needed to replace a test system, so the uniformity of the platform has enabled this step to be avoided with the design of Shell Sync. Rather than using the SWPM process, Shell Sync extracts all the data for the repository and client-independent data from the source system into a proprietary format, and then on the target system imports those files directly into the HANA database, replacing the existing content for the shell and simultaneously clearing down any client-dependent data. Once the process has completed, the user can then leverage Client Sync to populate the system with client-dependent data. Client 000 is also included completely in the Shell Sync copy, replacing what was previously stored for that client with data from the source.