Business Configuration (BC) Sets in Global Rollout Projects — Lessons Learned

Matthias Melich

Dr. Matthias Melich joined SAP in 1995. Since 1998, he has been responsible for Product Management of Customizing Tools. In January 2001, he joined the Solution Management Technology department, and is Manager of the Technical Product Integration team, which focuses on knowledge management, product planning, and development of implementation standards and guidelines.

(complete bio appears on page 136)

Starting with R/3 Release 4.6C, Business Configuration (BC) Set technology offers new functionality for customizing settings in global rollout projects that allow you to:

1. Define at headquarters the processes (i.e., customizing settings) that are standardized across all subsidiary units of your organization.

2. Put all customizing that belongs to the global template[1] into a virtual "box" within the R/3 system and ship it to subsidiaries. Each subsidiary then unpacks this box in its system and activates the content, thus coming into compliance with the standards set at headquarters. In addition to activating settings, BC Sets allow you to lock down settings that belong to global processes so that they can't be changed after they are rolled out.

My previous articles in this publication introduced BC Sets and explained the new functionality provided with Release 4.6.[2] (For a refresher, see the sidebar on the next page.) Since then, a number of SAP customers have either adopted this technology for their global rollout projects or are actively considering using it. The purpose of this article is to share the experiences of two early adopters of BC Sets, so that you can benefit from the lessons they learned. In these pages you will see:

[1] A *global template* defines the global requirements of an enterprise's SAP system. It includes customizing settings for core processes, reports, master data, and so forth.

[2] "Need to Get Your Customizing and Testing Ready for a Global Rollout? Use BC Sets and the Test Organizer to Smooth the Way" (March/April 2001) and "Evaluating, Incorporating, Altering, and Testing Headquarters-Defined Customizing: Lessons for Subsidiaries" (May/June 2001).

A BC Sets Refresher

A Business Configuration (BC) Set is a snapshot of customizing settings taken at a given moment in time (e.g., after a business function is tested successfully). This copy of the settings is stored in a container (the BC Set) that is separate from the original customizing tables. BC Set technology was introduced with Release 4.6C.

You can do more than just take a snapshot of customizing settings with BC Sets, however. SAP provides features that make BC Sets useful in global rollout scenarios:

☑ Attributes may be assigned to the customizing settings that are contained inside a BC Set. For example, you can assign the attribute “fix” to settings that are not supposed to be changed in the future, or you can assign the attribute “variable” to settings that may (or should) be replaced by other values. The attribute “variable” is especially important for settings that are specific to particular locales or organizational units.

☑ Customizing settings that have been captured in a BC Set can be applied to customizing tables. This process is called *activation*. During activation, the system copies the contents of a BC Set and writes the values back to the tables where these values belong.

☑ How SAP customers included BC Set creation, validation, shipment, activation, and maintenance in their global rollout methodologies

☑ The technical issues that caused the most headaches for our customers

☑ How to use the enhanced activation log (which was modified as a direct result of customer feedback) and the program created for evaluating the compatibility of any given IMG activity with BC Set functionality

Along the way, you’ll also learn the answers to these frequently asked questions:

- What is the fastest way to create BC Sets?
- When should BC Sets be created — during customizing or after customizing settings have been tested?
- How can I integrate the creation of BC Sets into my global template development process?
- Is there a recommended naming convention for BC Sets?
- How big can BC Sets be? How big *should* they be?
- How do I test BC Sets?
- How does “locking” work?
- Which BC Set functionality is available for a given IMG activity?
- Which support package do I need in order to have the best BC Set functionality?
- After rollout, how can I find out which BC Sets were activated and who activated them?
- Who should create BC Sets — all of my

The standard procedure for employing BC Sets in a global rollout is as follows:

1. The headquarters template team defines transport requests for those processes that will make up the global template.
2. Customizers at headquarters customize the template and save all settings to these requests.
3. The requests are converted to BC Sets using the BC Set maintenance transaction SCPR3.
4. The BC Set content is refined and enriched (i.e., values are assigned the attribute “fix” or “variable”).
5. The BC Sets (together with the other ingredients of the template) are put into transport requests and shipped to subsidiaries for implementation in one of two ways:
    - If a subsidiary is starting with a blank system, the BC Sets are imported and activated.
    - If a subsidiary is implementing the template in a running system, there is an intermediate step before activation: after importing the BC Sets, the subsidiary must compare their content against the local system in order to get a feel for the changes that will occur through the template implementation.

customizers or only a small expert team of customizers?

- What training resources are available for learning how to create, test, and activate BC Sets?

Let’s begin by meeting our two protagonists, DSM and BOC.

DSM and BOC — A Tale of Two Early Adopters

DSM and BOC were among the first companies to use BC Sets as part of a global template rollout project. The decisions they made, the methods they adopted, and the results they achieved were all products of each company’s individual makeup and system landscape. Let’s take a closer look at the details of each company’s project.

DSM’s Global Rollout Project

Here are some pertinent facts about the global rollout project at DSM[3]:

- **R/3 release**

  DSM chose R/3 Release 4.6C as the core release for the global template, using support package 20.

- **Common kernel**

  At DSM, the common kernel encompasses the following modules:

    - Sales and Distribution (SD)

[3] DSM is a highly integrated group of companies that is active worldwide in life science products, performance materials, polymers, and industrial chemicals. The group has annual sales of €8.1 billion and employs about 22,000 people at more than 200 sites worldwide. DSM’s strategic objective is to secure, by 2005, a place among the world’s leading specialty companies, with businesses characterized by high added value, strong growth, and more stable profit levels. DSM’s focus will be on advanced chemical and biotechnological products and performance materials.

  - Advanced Planner and Optimizer (APO) Demand Planning
  - Financial Accounting and Controlling (FI/CO)
  - Materials Management Purchasing (MM-PUR) during phase 2 (see below)

- **System landscape**

  The system landscape includes 19 SAP R/3 installations serving 4,000 users.

- **Project timeline and geographical scope**

  DSM's rollout project has two phases:

  - Phase 1: The Netherlands, Belgium, Germany, France, Spain, Sweden, Italy, and UK
  - Phase 2: US, Canada, Mexico, Turkey, Poland, Ireland, Australia, and Singapore

  The development of the common kernel for phase 1 was started and completed in 2001. At the beginning of 2002, this kernel was shipped to the development systems of DSM's business groups (called "subsidiaries" throughout this article for simplicity). The subsidiaries then started to enhance the kernel with local requirements. In May 2002, three subsidiaries went live (as scheduled). More followed throughout the year.

  The development of the phase 2 kernel (which enhances the phase 1 kernel) was completed in March 2002. The live date for this kernel version in the US is planned for March 2003.

- **Team roles**

  DSM's global template team consists of 100 people (60 consultants and 40 DSM employees). The BC Set team consists of three people who created the BC Sets at headquarters.

BOC's Global Rollout Project

Here are some pertinent facts about the global rollout project at BOC[4]:

- **R/3 release**

  BOC chose R/3 Release 4.6C as the template's core release. Initially, support package 19 was used; the support packages are applied every three months.

- **Common kernel**

  At BOC, the common kernel encompasses the following modules:

  - Financial Accounting and Controlling (FI/CO)
  - Materials Management (MM)
  - Sales and Distribution (SD)
  - Container Management (a BOC custom development for returnable containers)

- **System landscape**

  Currently, there are two template-based 4.6C production systems (Europe and Americas) with more than 5,000 users. There are three further systems (Africa, Asia, and South Pacific) to follow. There is a single development system.

- **Project timeline and geographical scope**

  The BOC template implementation is conducted by project within region:

[4] Serving 2 million customers in more than 50 countries, The BOC Group is one of the largest and most global of the world's industrial gases companies. It employs some 43,000 people and had annual sales of more than £4 billion in 2001. Three significant businesses have grown in parallel with BOC's industrial gases activities: BOC Edwards supplies ultra-high-purity gases and associated equipment to the semiconductor industry and is also well-known for its vacuum pumps. Gist is a specialist logistics company operating principally in the UK. The third is Afrox Hospitals based in South Africa — the largest supplier of private healthcare in Africa and one of the largest outside the US.

- Europe: UK and Ireland (live); Poland (in progress)
- Americas: US and Canada (live); will include South America
- Asia: Malaysia, Thailand, China, Taiwan, Philippines, and Korea (2003); will include Japan, India, Pakistan, and Bangladesh
- South Pacific: Will cover Australia and New Zealand
- Africa: South Africa; likely to include other countries

The pace of the deployment is influenced by each project business case.

The initial template build was done as part of the upgrade of the existing UK/Ireland system (3.0F to 4.6C), which took place from October 2000 to April 2001. The first rollout to North America was carried out from April 2001 to September 2002, with a phased implementation by module (FI/CO, then MM, then SD) and by country (Canada, then US).

- **Team roles**

BOC's project teams consist of between 30 and 80 people at any given time. Creation and maintenance of BC Sets is part of the role of the six-person template team.

Choosing a Global Template Strategy

Before diving into a global template project that incorporates BC Sets, it's important to take some time to construct a rollout strategy. Knowing how you want to deal with the inevitable differences that exist among subsidiaries while rolling out a standardized core system will shape how you construct the global template. For example, will you allow your subsidiaries to have their own development (DEV) boxes,[5] or will you allow them to have quality assurance (QA) and production (PRD) systems only? The architecture of the global template, the global and local system landscape, your attitude toward local changes to the template, and so forth — all of these considerations will shape your decisions about BC Sets.

A global template contains the following components:

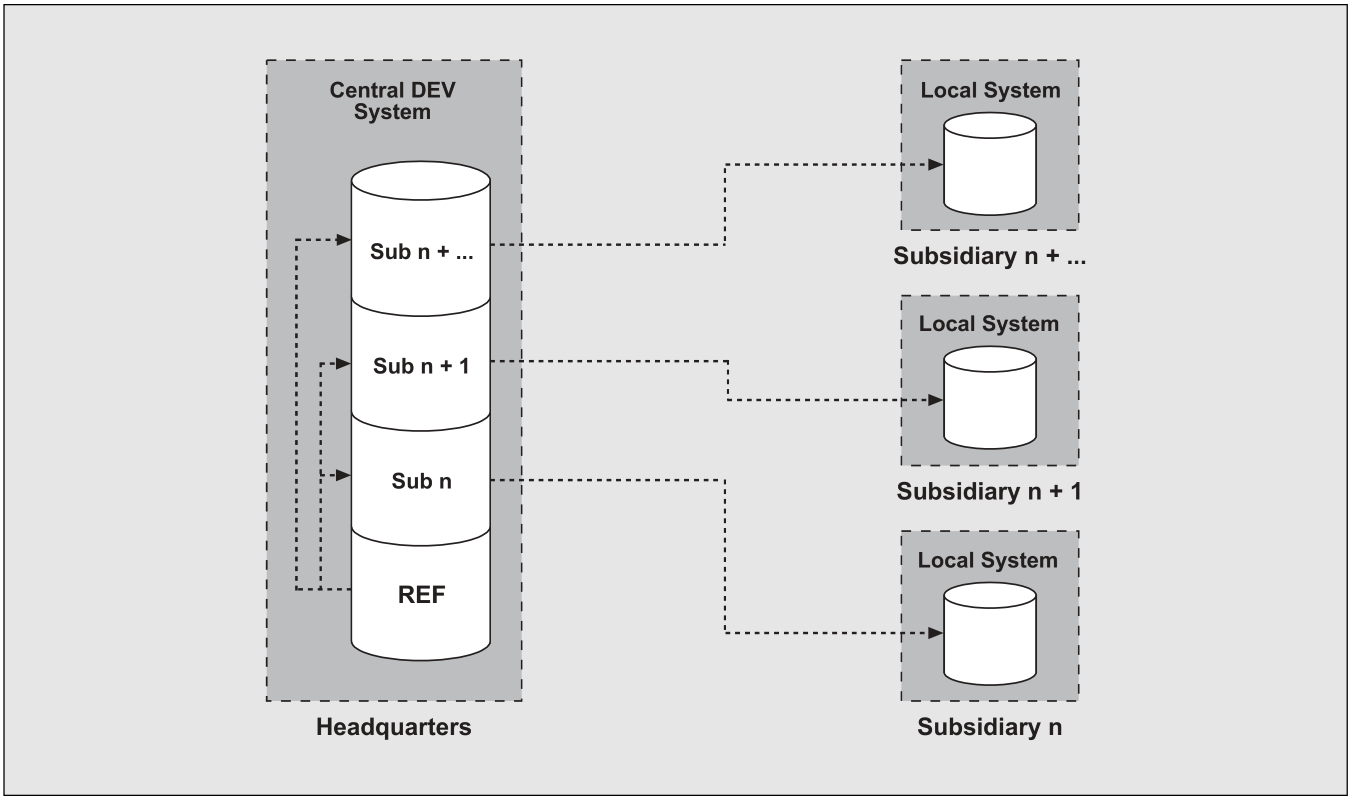
- A kernel, consisting of:
    - Reports, master data, and so forth
    - Customizing settings that describe the core processes of the company (i.e., the processes that are supposed to run at every subsidiary installation)

[5] At DSM, each business group has its own DEV system, which makes it easy for a group to deviate from the global template.

✓ ***Note!***

The role of the SAP BC Set team during DSM and BOC's global template projects was to review the proposed usage scenarios of BC Sets with respect to technical feasibility. In other words, each company told us how they planned to use BC Sets and we told them whether their ideas could be realized from a technical point of view. In all other respects, the DSM and BOC projects were completely independent endeavors. All decisions regarding these projects were made solely by the customers and their consultants. That said, the usage scenarios for BC Sets described in this article are certainly valid; I encourage you to consider them seriously if you decide that they make sense in your own system environment.

Figure 1 *Global Template Architecture*



- Transport request files

- Test descriptions and plans, so that subsidiaries can verify that everything works as intended once the template has been applied and also ensure that all settings have been made

- Instructions and learning material

In most cases, the global template kernel cannot and does not cover all the requirements of each subsidiary unit of a company. At DSM and BOC, for example, it was clear from the beginning that each subsidiary would need region-specific and business-unit-specific additions to the company's common SAP kernel. Both companies addressed this need by adopting a strategy involving the use of several different clients in a central system, including:

- A reference (REF) client to store the template, which contains the kernel that is standard across all business units.

- One client for each of the subsidiary units. Each of these clients is built by activating the customizing of the kernel (template) and then adding the localizations required for the appropriate region or business unit. The client is then shipped to the subsidiary's local system.

Figure 1 illustrates this strategy, showing the template development system at company headquarters and the subsequent rollout. The client kernels (*Sub n*, *Sub n + 1*, and *Sub n + …*) are rolled out to their respective subsidiaries; the REF client is retained at headquarters.

While DSM and BOC follow the same general strategy, they differ with respect to the internal structure of the kernel. Whereas DSM sees the kernel as

Why Use a Global Template?

Project members of the global rollout teams at our case-study companies gave these reasons for using a global template to roll out R/3 to subsidiary units:

☑ **Transfer of best operating and commercial practices:** The knowledge transfer of “best practices” from headquarters to subsidiaries can be facilitated by incorporating the expertise and SAP know-how that exists at headquarters into a global template that is shipped to all business groups or regions. (A global template can contain descriptions and instructions in addition to software, settings, and data; the customizing and reports it contains can also be part of a company’s best practices.)

☑ **Launch of a worldwide e-business strategy:** Implementing an e-business strategy requires the transfer of knowledge (such as goals, timing, and high-level descriptions of how goals will be reached) as well as software components. A global rollout that reaches all subsidiaries provides a perfect vehicle for this transfer, as the information and software to be transferred can be woven into the template (especially if the company selects mySAP.com as its e-business platform; BC Sets can be used with all mySAP components based on SAP Web Application Server 6.10).

☑ **Integrated supply chain management and optimization:** The introduction of a core system that is eventually established worldwide provides a means for setting up an integrated and optimized supply chain (among subsidiaries or suppliers).

☑ **Providing the basis for potential shared services among subsidiaries:** Subsidiaries can benefit from services that are provided by corporate headquarters (for example, system

(continued on next page)

a solid block, consisting of a core system and set of processes that must be accepted in its entirety by each subsidiary, BOC makes a distinction between the core processes that every subsidiary uses and what they call *configurable extensions* — i.e., variations to the kernel that a subsidiary can choose to use, but are not required.[6]

Another difference between the DSM and BOC template strategies is the starting point for the global template kernel. DSM chose to build an entirely new kernel; BOC decided to use its existing UK client as the starting point for its global kernel. (See the sidebar above for more on the decision to use a global template.)

Choosing a BC Set Strategy

With the global template strategy settled, you must next decide how you will prepare the customizing settings that will be in the template. Which settings will be in transport requests and which in BC Sets? (Not all process activities are good candidates for BC Sets, as you will see.) Which method will you use to create BC Sets? You have two choices:

- Conversion of transport requests using either:

[6] Configurable extensions were critical in helping the global template gain acceptance among BOC subsidiaries, since these extensions provided them with more freedom.

(continued from previous page)

maintenance and template maintenance) instead of developing their own services or ordering them from outside the company.

☑ **Cost savings:** Using a global template can result in an enormous cost savings. For example, BOC estimates that the template approach will reduce the implementation costs of their worldwide rollout by £15 million compared to independent local systems (the savings come principally through reuse of configuration, programs, training, and documentation).

☑ **Reduction of the ongoing costs at the offices of subsidiaries:** Cost reduction is often the main argument for implementing a global template. BOC, for instance, expects that its global template will save £7 million per year. Much of this comes from standardization/reuse and from having a single development system for upgrades, support packages, and enhancements.

☑ **Cost savings associated with reduction of the system landscape:** Subsidiaries that have a full landscape consisting of DEV, QA, and PRD systems have a lot of freedom in terms of developing new tools and customizing. That freedom may cause problems in global rollouts because it allows local teams to easily deviate from the global template. However, if you take DEV systems away from subsidiaries, create all customizing centrally, and then roll it out via a template containing BC Sets, you can both reduce the system landscape and save money in the following ways:

- You do not have to pay for DEV system maintenance.
- You minimize the risks of template deviations in the subsidiaries.
- Fewer deviations from the template mean fewer problems with upgrades of the template and the SAP release, and thus fewer costs associated with servicing the SAP release and the template.

  - Immediate conversion
  - Delayed conversion
- Creation of BC Sets using the IMG

DSM's method was to convert transport requests; BOC's was to use the IMG. Let's take a closer look at the two different methods in practice.

DSM's Experience

DSM planned to create BC Sets in this way:

- **Method (create BC Sets from transport requests):** DSM decided to convert the transport requests in which all of the customizing that belongs to the template is recorded. The main argument for this decision was that this method would save time because it would involve going through the IMG only once (to create the transports) instead of twice (first to customize the system and then to create the BC Sets). Actually, it seemed to DSM that even more time could be saved since the transports already existed and could be "reused" in the template project.
- **Timing (immediate conversion of transport requests):** For each process, DSM planned to immediately create a BC Set upon completing the customizing in the development system. They reasoned that by using this method they could fully integrate BC Set creation with the customizing process. This

method is known as *automatic recording of customizing values*.

In the course of creating BC Sets from transport requests, DSM encountered a problem, however: the BC Sets did not always contain all of the data stored in the corresponding transport. The reason for this behavior lies in the IMG architecture, which determines what goes into a BC Set and what does not (whether you are creating BC Sets from the IMG or from transports). The majority of activities in the IMG allow users to manipulate data that is stored in tables — and these activities can be converted into BC Sets easily. There are some activities, however, that contain a "trigger" to start system actions, such as the generation of a report based on customizer entries. Each report is a repository object and is therefore stored in a transport of type "syst." However, BC Sets can only handle customizing data, not repository objects, and customizing is stored in transports of type "cust." The BC Set conversion utility can therefore only handle transports of type "cust."

DSM accepted this behavior of the conversion tool, but requested that our development team provide a way to find out, for a given IMG activity, whether it can be used with BC Sets or not. The answer to this requirement was a report — nicknamed "Rupp's Program" — which I describe in detail in a later section called "Enhanced BC Set Functionality." Briefly, Rupp's Program will tell you, for any given IMG activity, the BC Set functions with which that activity is compatible.

The trigger mechanism is also the major reason for another effect that DSM discovered. If you enter parameters in the IMG and then use the BC Set maintenance transaction (SCPR3) to create a BC Set from the IMG activity, and then finally compare this BC Set to one that was created from the transport that recorded the customizing, you will (sometimes) find differences. For example, the BC Set created using a transport request may contain *more* data (i.e., additional data that you never entered) than the one created using transaction SCPR3. The reason for this seemingly strange behavior is, again, that some activities contain a trigger that creates additional customizing values based on the input provided by the customizer. When these values are captured by a transport request that is, in turn, converted to a BC Set, these values obviously show up in the converted BC Set. If you go into transaction SCPR3 to create a BC Set for this activity, however, you need to know which additional values you have to include. The problem is that often the BC Set creator is unaware of this mechanism and will therefore not include these values — with the result that the manually created BC Set is incomplete. If you were to activate such a BC Set, the data would be missing in the target system. Rupp's Program also addresses this problem.

✓ Recommendation

Use conversion from transports as your default method for creating BC Sets in a template project. This way, your BC Sets will contain all possible customizing values for an activity, including activities that have a trigger to create additional values based on user input.

Converting transport requests into BC Sets during the customizing phase (the immediate conversion method) proved to be difficult. DSM had expected that an existing BC Set could be updated using a transport request, and had anticipated that they could keep their BC Sets in synch with the template during the customizing phase. What DSM did not know was that every time a transport request is converted, a new BC Set is created — you cannot open an existing BC Set and update its contents by pulling in the information that resides in the corresponding request. If you have to re-create your BC Sets each time you update your customizing, then you either must have a clear procedure established for deleting the outdated BC Set before starting the conversion utility, or you must devise a naming convention that can handle these "versions." The problem with relying on a naming convention, however, is that you can never tell by looking at a given BC Set whether or not it

Naming Conventions for BC Sets

In one of my previous articles,* I described a naming convention for BC Sets that was used successfully by SAP's best practices department. Briefly, this convention is as follows:

<SAP component>-<process name>-<version number (in increments of 10)>

So, for example, a BC Set might be named APO-VMI-10 where "APO" designates the Advanced Planner and Optimizer component, "VMI" stands for the Vendor-Managed Inventory process, and "10" indicates that an earlier version (version 00) exists. The benefits of this convention are that you can see which component the BC Set is for, which process it describes, and which version it is.

However, there is no such thing as a perfect naming convention, because projects are highly individual. For example, DSM wanted to be able to tell the following from a BC Set name:

- Which process the BC Set defines
- Whether these issues were reported during the creation, testing, or activation process, and if so, what the issue number was
- For which template version the BC Set was created
- The relationship between the BC Set and the transport request used to create it

DSM changed the recommended convention by coding the following information into its BC Set names:

- **Process name:** During their template planning process, DSM decided to use a reference scheme for the business scenarios included in their common core. This reference scheme assigns a unique

* "Need to Get Your Customizing and Testing Ready for a Global Rollout? Use BC Sets and the Test Organizer to Smooth the Way!" (March/April 2001).

reflects the current state of template development. (For recommendations on naming conventions, see the sidebar above.)

For this reason, DSM decided to postpone the creation of BC Sets until after all processes were tested successfully, when the customizing of the core processes would be stable (the delayed conversion method). They reserved a two-week period in their project plan for BC Set creation. Of course, the downside of this approach was that the project timeline had to be extended.

With an additional phase added to their project plan for BC Set creation, DSM was naturally interested in a fast conversion of the existing transport requests. Because the BC Set conversion utility only converts transport requests to BC Sets one at a time, DSM decided to write a report that would allow them to convert many requests to BC Sets simultaneously.

DSM also wanted a utility for changing all field attributes in a BC Set. When you create a BC Set (either from the IMG or from a transport request), the fields always carry the attribute "default." DSM wanted a different default ("fix") for field attributes in BC Sets. Unfortunately, the BC Set maintenance transaction (SCPR3) does not allow you to change all attributes in a given BC Set (and the conversion utility does not offer this functionality either). So DSM included this functionality in their mass conversion report.

number to each business scenario (and the processes that it includes). This scheme was provided by their consulting partner.

- **Issue number:** If an issue was filed regarding a BC Set during the creation and activation process, and if the solution of this issue required creating a new BC Set, then the issue number was included in the BC Set's name as well.
- **Release number:** DSM wanted to be able to reference the template release.

In addition, DSM indicated the transport request used to create the BC Set by using the same naming convention for transports and BC Sets. As their rollout focused only on R/3, they left out the component name used in the SAP naming convention.

So, using the DSM-enhanced naming convention, a BC Set might be named VMI-15-02 where "VMI" is the process name, "15" is the issue number, and "02" is the release number.

While there is no single, recommended naming convention, I do strongly recommend that you have at least a *clear* naming convention. The name should contain an indicator that the BC Set is a customer BC Set (use the "Z" or "Y" prefix, or a customer namespace such as "/EXAMPLE/") and a meaningful hierarchy of abbreviations or numbers that probably will have to be determined on a project-by-project basis. It is also a good idea to code issue numbers into the name of a BC Set. For example:

- **/SPJ/_CRM_FSD_SCE_V10:** Customer SPJ's BC Set for Customer Relationship Management (CRM), in the Field Scenario and Dispatch (FSD) hierarchy, which is a scenario (SCE), and is version 10.
- **/SPJ/_CRM_FSD_HS_PRO_V10:** Again, we are in the same hierarchy (FSD), one level further down in Handheld Services (HS), which is a process (PRO), and is version 10.
- **/SPJ/_CRM_FSD_HS _USCI_PRS_V20:** We are further down in the FSD hierarchy in the Update Service Cost Information (USCI) process step (PRS), which is an updated version (20).

The SAP BC Set development team is currently evaluating whether this functionality is something many, or even all, customers want. If it is, we will include it in our planned features for performing mass conversions and changing all attributes in a BC Set.

BOC's Experience

BOC took a different approach to the creation of BC Sets. They wanted to use their existing UK client kernel as the basis for their template, but the transport requests that were used to create the UK client were no longer available, as they originated from a 3.0F system. BOC was therefore looking for a fast and cost-efficient way to reuse the existing customizing data in the UK client. At the outset of their project, they evaluated two options:

- **Option 1:** Make a copy of the existing UK client; strip it down manually until it contains global customizing settings only (i.e., settings that are part of the common kernel defined by BOC); and then start enhancing this client until it contains the full global template. This option does not involve using BC Sets at all.
- **Option 2:** Use BC Sets to capture those portions of the UK client that are relevant for the template (using the IMG method of creating BC Sets) and later transfer those customizing settings into the template client via activation.

Why BC Sets?

I have discussed the benefits of BC Sets at length in previous articles. Here are the particular reasons DSM and BOC gave for using this technology in their global rollout projects:

☑ **Protection of the kernel's customizing:** BC Sets allow you to lock settings in subsidiaries' systems. From headquarters' point of view, this functionality is very attractive because it guarantees that a certain set of business processes runs in all systems of the company.

☑ **Ability to have a complete change history:** Being able to take a snapshot of customizing settings, and having those settings stored in a container separate from the customizing tables, makes it easy to keep track of customizing changes of the template. If you take a snapshot every time a core process has to be adapted, then you have a complete change history.

☑ **Transparency:** If you capture your template customizing in BC Sets, it becomes more apparent what is part of the template and what is not, for the following reasons:

- You can show which IMG activities are used in a template.
- You can display which parameters of a given IMG activity are part of the template.
- You can evaluate which attributes the parameters carry ("fix" or "default").

From earlier experiences, BOC knew that stripping the client down and then enhancing it would take about six weeks. After evaluating the BC Set maintenance transaction (SCPR3), BOC chose to take this path and use BC Sets instead, because — in addition to other advantages (see the sidebar above) — they expected it to be faster than stripping the client.

Lessons Learned

☑ **Wait until customizing is completed before you start creating BC Sets.** As DSM discovered, updating BC Sets via transport requests is not possible with the current functionality. You should therefore not create a BC Set until the customizing of a process is sufficiently tested.

☑ **Reserve a phase in your project plan for creating BC Sets.** As the experience of DSM demonstrates, BC Sets require their own phase in a global rollout project. The extra time they require means that they are an additional cost-driver, but the costs are more than recovered. (For the cost-saving benefits associated with BC Sets, see the sidebar on pages 117-118.)

☑ **If you want to reuse an existing client for your template,** use BC Sets instead of copying the client and stripping it down. As BOC found, it will save many weeks of customizing and is more suitable for adding additional content at a later time.

Determining BC Set Size

Both DSM and BOC were faced with the question of BC Set size: Is it better to create big BC Sets that include customizing for a complete process? Or is it

☑ **Business process orientation:** Customizing that is captured in BC Sets can be grouped according to business processes, allowing you to align the customizing settings with business requirements.

☑ **Reuse/transfer:** Among the major challenges of a rollout are how to reuse existing knowledge or system parameters and how to transfer them into the template, as well as to the subsidiaries. Especially in the area of customizing, you often have to "move" a process from one system to another, which means isolating the customizing parameters that define this process and loading them into another system. BC Sets are perfect for this task because they allow you to take snapshots of existing customizing and easily move these snapshots (either via transports or upload/download) into other systems.

☑ **Maximum flexibility:** An important aspect of the design of a template is its flexibility. A well-designed template will establish a method for dealing with local information that needs to be included in global procedures. BC Sets offer an elegant mechanism — variable fields — that allows headquarters to define the customizing of a process while leaving room for country-specific or business-unit-specific entries.

☑ **Generic approach:** The flexibility of BC Sets offers another advantage — if you can predefine areas in your template where subsidiaries can include their country-specific or business-unit-specific parameters, your template then shrinks, because it only needs to account for generic aspects of the global processes. The concept of variable fields provides headquarters with the technological key to highly generic templates.

better to create smaller BC Sets (on the IMG activity level, for instance) that can then be aggregated to provide a complete definition of customizing settings for a process? This aggregation of BC Sets is called a *BC Set hierarchy*. A hierarchical BC Set, which may contain one or more BC Sets, allows you to structure BC Sets according to process.

DSM's Experience

DSM initially wanted to build large BC Sets, but ran into sizing issues during testing of the functionality. The main problem they encountered, and the earliest, was that the system timed-out when converting transport requests that contained a lot of customizing data. DSM reported this problem to our development team via OSS. Although we provided a fix to speed up the conversion (which is included in Basis support package 36 for 4.6C), the conversion of large transports may still cause problems.

DSM then investigated whether it was possible to split large transports into smaller units that could be converted. The problem with splitting a large transport, however, is that you have to know the exact order in which the individual objects in the request need to be put into the system. If you have three objects (X, Y, and Z) that need to enter the system in alphabetical order, and you put X and Z in BC Set 1, and Y in BC Set 2, then the activation of either one leaves you with an inconsistent customizing state. Even the back-to-back activation of BC Sets 1 and 2 is only consistent *after* the activation of BC Set 2 — the intermediate customizing state after the activation of BC Set 1 produces an inconsistent system state.

To avoid this problem, DSM decided to change its BC Set sizing policy dramatically by imposing the following rule:

1 IMG activity = 1 transport = 1 BC Set

In other words, for each process, DSM did the following:

1. Created one transport request for each IMG activity involved in the process.

2. Converted each of these transport requests into a BC Set.

Unfortunately, one customizing team started configuring its process — Order to Cash (OTC) — before this procedure was established. As a result, the OTC BC Set contained 400,000 key lines, and took about 30 minutes to load and 60-90 minutes to activate (in a system without a lot of other traffic).

BOC's Experience

BOC wanted to use its UK client as the basis for its global template. It was clear that the fastest way to capture this knowledge was to develop large BC Sets to contain the customizing settings for a specific business process.

However, like DSM, BOC discovered the downsides to using large BC Sets. In the future, they will follow a policy of creating small BC Sets with only a few IMG activities in each and using hierarchies to combine them for activation. BOC will, however, continue to build BC Sets from IMG activities rather than transport requests.

Lessons Learned

☑ **Small is beautiful.** As a rule-of-thumb, BC Sets should not contain data from many activities. However, it's not the *number* of activities that matters, but the *amount of customizing data* that you include in a BC Set. There are no explicit limits for the amount of data a BC Set can contain. How much data depends on factors such as the layout of the customizing table.

☑ **Use hierarchical BC Sets.** This is corollary to the "small is beautiful" rule. Instead of creating a large BC Set that contains all the customizing data for a process, create smaller ones — one per IMG activity — and group them together using hierarchical BC Sets.

☑ **Establish clear rules for creating BC Sets before allowing teams to start.** DSM learned this rule the hard way by allowing one team to develop an OTC BC Set before the central BC Set team had completed testing the functionality of the BC Set. In the end, DSM decided to rework this BC Set completely, creating small BC Sets for the IMG activities involved in OTC.

☑ **Prepare your customizers for changes.** If you decide to create BC Sets by converting transport requests, you will have to convince customizers to create "small" transports (i.e., transports that capture the customizing of one or only a few activities). Alternatively, you can create the requests beforehand and hand the customizers a list that tells them which activity is saved to which request. This may not be the way your customizers typically work. If you decide on the BC Set maintenance transaction (SCPR3) as your method for creating BC Sets, be sure to prepare your customizers for the difference in how the maintenance transaction UI presents customizing parameters (as compared with the IMG UI). The BC Set maintenance transaction focuses on *customizing objects* whereas the IMG focuses on *IMG dialogs*.

Testing BC Sets

Testing the content of BC Sets is crucial to the success of the complete template. You have to make sure that:

- All customizing that belongs to the global template is contained within the BC Sets.
- The BC Sets can be activated properly in the subsidiaries.
- All transports have been converted.

Testing BC Sets is a major task of the central BC Set team in your global rollout project.

DSM's Experience

DSM implemented the following procedure to unit test BC Sets created from transport requests:

1. **Content-completeness check:** The customizer who customized the activity included in the BC Set calls up the BC Set maintenance transaction (SCPR3) and verifies the content. If he or she encounters a problem, the content is readjusted at headquarters by the BC Set creation team. Remember what I said earlier about triggers? BC Sets that contain triggers do not pass this test, because the customizers will see values in their BC Sets that they never entered. This issue can only be resolved by the central team using Rupp's Program (more on this later).

2. **Activation check:** For each BC Set that passes step 1, the central BC Set team starts an "activation simulation" run in transaction SCPR3.[7] If this simulation fails, the BC Set is readjusted.

3. **Post-activation check:** The central BC Set team activates the BC Set in a test client of the reference DEV system and corrects any errors.

After a BC Set reaches step 3, it is considered okay from a technical standpoint. The BC Set's content is checked implicitly when the global business processes are tested during sequence testing. If a process does not work as desired and if the source of this mistake is a BC Set, then the BC Set is fixed and has to undergo the three-step testing procedure again.

The last implicit check is the integration test. Here, the same procedure is applied that is used during the sequence testing — faulty BC Sets return to the source and unit tests are performed again.

> ✓ ***Note!***
>
> *After BC Sets have been created, be sure to update your BC Sets anytime you change customizing, and remember that each BC Set change must then be unit tested as described in steps 1-3.*

BOC's Experience

Like DSM, BOC had a small team that first verified the contents of BC Sets. Then they activated the BC Sets and checked again to see whether the customizing had activated fully. Process testing is part of BOC's deployment project once local configuration and data take-on is complete. Any global configuration problems discovered at this stage are then fixed and updated in BC Sets separately.

Lessons Learned

☑ **Remember to update.** Update affected BC Sets if customizing changes are necessary during sequence and integration testing.

☑ **Set aside a test client.** Be sure to reserve a test client for BC Set activation tests during unit testing.

☑ **Have a plan.** Establish clear procedures for how BC Sets are fixed during sequence and integration testing.

Global Rollout — BC Set Activation and Locking

The activation feature of BC Set technology is a mechanism for copying BC Set content into the customizing tables of an SAP system.

Activation does two things:

[7] Starting with support package 37 for 4.6C, the simulation is run implicitly. Select the option "Cancel with log" on the activation pop-up.

1. It creates customizing settings in the subsidiary system. (The BC Set content that was captured at headquarters is copied into the customizing tables of the subsidiary system.)

2. It applies locks (where possible).

Like most customers in the global rollout arena, DSM and BOC were intrigued by the locking functionality of BC Sets, which allows the global template team to "lock" customizing settings that are not supposed to be changed by the local units. Here's how locking works:

1. You set the attribute "fix" for all parameters that must not be changed by your organization's business units and/or countries.

2. You activate the BC Set in the subsidiary system.

3. The activation procedure applies the attribute "fix" to the appropriate system settings (the same ones that are attributed as fixed in the BC Set).

Before DSM and BOC made their respective decisions about whether to make use of the locking mechanism, they tested these functionalities thoroughly and found the following issues:

- **The activation log was insufficient.** The activation log that DSM and BOC were using consists of a screen that lists, during the activation process, which views and tables have successfully received data and which have not.[8] The main criticisms were that:

    - The information logged was insufficient.

    - The information was transient. The activation log is not stored in the system, so it is gone if you leave the log screen. This meant that DSM and BOC had to print out the log screen in order to have a record of what happened during the activation process.

- **Not all customizing entries marked as fixed in the BC Sets were locked after activation in the subsidiary system.** The locking mechanism does not work everywhere in the IMG. The majority of customizing activities rely on generic customizing dialogs, and these work fine with respect to locking. There are customizing activities that rely on modified customizing dialogs or on programmed dialogs, however, and in these cases the locking mechanism does not work.

Both DSM and BOC wanted an activation log that would remain in the database and be permanently available for review. They wanted the activation log to show:

- Which BC Set was activated

- Who started the activation

- When the activation was completed

- Which values were supplied for the variable fields

All of these features are now included in the new activation log provided with support package 35 for 4.6C. I explain how the new activation log works later, in the section "Enhanced BC Set Functionality."

As for locking, DSM and BOC requested a simple means for finding out where locking works and where it doesn't. This functionality is provided in the report (Rupp's Program) that evaluates the compatibility of IMG activities with BC Set functions (this is also explained in the section "Enhanced BC Set Functionality").

DSM's Experience

DSM chose not to use activation as the main method for creating customizing settings in the systems of their subsidiaries. Instead, they:

1. Imported the transport requests that were used as the basis for their BC Sets into the systems of their subsidiaries.

[8] DSM and BOC were working in Release 4.6C with support packages 20 and 19, respectively.

2. Put all the BC Sets that corresponded to these requests into transport requests (keep in mind that these were easy to find because of the identical names of requests and BC Sets, as explained in the sidebar on pages 120-121) and imported them into the systems of their subsidiaries.

3. Activated the BC Sets to apply locks.

With this method, the standard customizing enters the system twice — first with the transports and then again during activation. The activation procedure overwrites the settings that came in with the transports with identical values, and locks are applied during activation. This method ensures that there are no problems with activation. (Activation is the BC Set function that carries the highest risk, because if a BC Set fails to activate correctly for any reason, the missing customizing can cause a process to run improperly.) In this way, even if activation fails, the customizing that would have been written with this BC Set is already in the system. The only issue then is that no locks are applied for this BC Set. DSM thus managed to find a way to separate the reuse of customizing from the application of locks. Risks are further minimized because no new customizing is written at the subsidiary sites.

One disadvantage I saw with this method, however, is that you lose the cost savings associated with BC Set reuse. Another disadvantage is that this method presupposes that the complete global organizational structure is modeled in the reference system. It forfeits the cost-saving advantage of having only a small organizational structure in the reference system that can be completed in the subsidiaries' systems via variable fields.

BOC's Experience

BOC decided to use the full activation functionality of BC Sets because, for them, activation had been identified as the means for more quickly and cost-effectively creating their global template client and their subsidiary clients. Rather than copying a client and stripping everything they did not want, they could simply use activation to quickly and easily apply their BC Sets to a new copy of client 000 (after creating the BC Sets in the source client).

To implement their global rollout, BOC:

- Used activation as the method for moving customizing into the subsidiary client (within the single development system).

- Used variable fields because they did not want to create the full company model beforehand. Instead, they wanted to allow subsidiaries to adjust business processes to their local organizational structures.

- Did not use the "fix" attribute but documented which configuration was global in the IMG. This method allows changes, where necessary, in the local development client.

Because this method made it crucial to know where BC Set activation did not work, BOC ran their usual tests to find out where functionality was not working properly, and then went to the customizing tables and fixed these problems manually.

Lessons Learned

☑ **For local activations, train the local teams.** Whether the teams doing this are local or central/global will depend on your organization. In either case, if you want to unleash the full potential that BC Sets offer in a global rollout, you have to ensure that the appropriate teams know:

- What a BC Set is

- How to compare BC Sets using the Customizing Cross-System Viewer (CCSV)[9]

- How to interpret the results of a BC Set comparison run

[9] For more on the CCSV, see "Enhancements in Customizing: Business Configuration Sets, the Customizing Cross-System Viewer, and the Activity Log" (November/December 1999).

Locking Should Not Be Your Only Reason for Using BC Sets

Like many (perhaps even all) customers in the global rollout arena, DSM and BOC were intrigued by the possibility of using BC Sets to lock those customizing settings that belong to the "hard" core of the template.

In my numerous conversations with customers, I usually see two reactions when I present the locking functionality — excitement and concern. Those representing headquarters are usually excited, because they don't want local business units to tamper with headquarters' definitions. By contrast, those from the subsidiaries are usually concerned: Will they lose their ability to override customizing settings from headquarters? What happens to their system if headquarters makes a mistake? Will they have to wait for a bug fix?

My response to these reactions is simple: Locking is an important feature, but it should not be the only reason, or even the main reason, for BC Set use in a global rollout project. Why? Because for technical reasons, some settings in the IMG cannot be locked. Keep in mind, though, that with the Customizing Cross-System Viewer (CCSV) you can always and at any time check a subsidiary's compliance with the template. Although you cannot *prevent* changes with the CCSV, you can find out who changed what and when, and remedy any violations. DSM successfully operates their template in their subsidiaries although they cannot lock down 100% of the settings they may have wanted to lock. BOC does not lock the configuration at all. Also, keep in mind that without BC Sets, you cannot lock settings at all.

- Which procedures to follow if template violations are found (either eliminate the violations locally or report them to headquarters with a request to change the template with the next release)
- How the activation functionality works
- How variable fields are handled during activations
- How to analyze the activation log
- Which procedures to follow if the activation procedure reports errors

☑ **Locking should not be the only reason you use BC Sets in a global rollout.** Not all IMG activities lock. See the sidebar above for more detail on this.

☑ **Test the activation.** Be sure to test the template activation before you ship your global template to subsidiaries.

☑ **Minimize the amount of customizing that you want to lock.** Ask yourself if everything that is global *really* needs to be locked. Train your local teams to use the CCSV to determine whether they will be violating the global template *before* they begin making changes to customizing settings.

Final Stats

Although every company's experience is different, you may find it helpful to know some of the "final statistics" relating to the DSM and BOC global template projects.

DSM's Results

DSM's template contains 250 BC Sets that cover approximately 2,000 IMG activities.

To create the BC Sets for phase 1 of the global rollout project (rolling out the template to subsidiaries in eight countries), DSM needed approximately two weeks to complete the work, including the conversion from transports, checking the content, assigning attributes, and activation. For the BC Sets required for phase 2 (the remaining subsidiaries in eight different countries), DSM needed less than a week. The second phase was an extension of the work of phase 1 and required no modifications to phase 1 BC Sets.

BOC's Results

BOC created 53 BC Sets for its template. These BC Sets include 582 IMG activities.

Building the BC Sets took 60 person-days of effort over a period of six weeks. Creating a new client now takes BOC less than three days using BC Set activation, as compared to the six weeks it took previously.

Enhanced BC Set Functionality

In this section, I share how SAP's customizing team enhanced the functionality of BC Sets as a result of customer experiences like DSM and BOC's, and I show you how to use this enhanced functionality. As mentioned earlier, we created a new report for analyzing IMG activities and learning which ones are good candidates for creating BC Sets. We also enhanced the activation log to address the issues that DSM and BOC discovered. (To take advantage of this enhanced functionality, you will need Basis support package 37 for 4.6C or higher.[10])

Rupp's Program: Analyzing the IMG

Both DSM and BOC found that it is important to know, for a given IMG activity, whether or not the activity can be encapsulated in a BC Set for use in a global template project. The term "use" has different implications depending on how you employ BC Sets:

- If you want to use a BC Set mainly for reference purposes, you must be able to create the BC Set error-free. The main purpose of creating a BC Set for documentation is so that you can record changes and compare them against the original tables. In other words, data that you enter during customizing must be in the BC Set.

- If you want to reuse BC Sets in the systems of your business units or subsidiaries, you have to be able to activate a BC Set without errors.

- If you want to prevent your subsidiaries from changing global customizing settings, you must be able to apply locking.

So, for each IMG activity for which you want to create a BC Set, you need to know if it is possible to:

- Create a BC Set (to document the activity)

- Activate the BC Set that is created

- Lock the customizing settings in the BC Set[11]

The "IMG Check of BC Set-Compatibility" report (RS_IMG_BC_SET_CHECK) — i.e., "Rupp's Program"[12] — helps you identify which

[10] For a complete listing of all support packages for 4.6C, 4.6D, 6.10, and 6.20 visit **http://service.sap.com/bcsets**.

[11] It is clear that locking can only work if activation works, and that activation can only work if you can create a BC Set in the first place. There will be more activities for which you can *create* a BC Set (but not activate or lock it) than there will be activities for which you can *activate* a BC Set. By the same token, there are more activities for which you can *activate* a BC Set (but not lock it) than there are activities for which *locking* of the BC Set is possible.

[12] Named after Tobias Rupp, the programmer who designed and developed the report. Given the original names, it's no wonder that the name "Rupp's Program" has stuck!

IMG activities can be used with BC Sets, and even the BC Set scenarios in which the IMG activity might be used. It tells you what can be documented, what can be activated, and what can be locked. This report was first used internally at SAP. Owing to the great demand for the functionality that it provides, we decided to make it publicly available with support package 19 for Release 4.6C. It is not available via transaction codes or menus in Release 4.6C. In Release 6.10, you can access the report with transaction code SCPRUPP.[13]

To access Rupp’s Program in 4.6C, open transaction SE38, in the “Program” field enter the report’s name (“RS_IMG_BC_SET_CHECK”), and then select “Execute” (click or press F8).

Figure 2 shows you the start screen of Rupp’s Program. Here, you can choose to either:

- Display the results of the program’s last run (“Display Last Check Result”)
- Run the program without displaying the results (“New Check Without Display”)

The date of the last run is displayed in the upper right corner (in Figure 2, this is “25.07.2001”). If this is the first time you are accessing Rupp’s Program, no date will be displayed (to be more precise, the display will read “00.00.0000”).

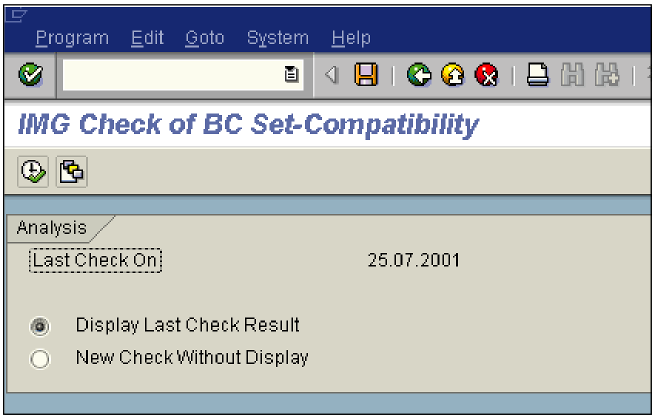
When Rupp’s Program runs, it checks the activities of the reference IMG with respect to BC Set compatibility (i.e., whether it is possible to create a BC Set for the activity, activate it, or lock it).

I suggest running the program once as a background job (which will take approximately 15 minutes) by selecting “New Check Without Display”; once you have the results, you can select the first option (“Display Last Check Result”). Subsequently, a new run will only be necessary when changes are made that affect customizing (i.e., when changes are made in the IMG structure or in the system itself — for example, when support packages are installed).

After setting the option on this screen to “Display Last Check Result,” choose “Execute” (click or press F8) and the report screen will be displayed. **Figure 3** shows you what this screen looks like when it opens. The IMG structure is displayed on the left. You can navigate to the relevant node and then expand the tree structure to see the IMG activities contained in that node. Expand those activities in turn until you reach the activity for which you want to see the results of the analysis. In **Figure 4**, you can see that I expanded the “Materials Management” node and then the “Purchasing” node. Finally, I expanded the “Create Purchasing Groups” activity in order to have a look at the customizing object “V_024.”

On the right side of Figure 4 you can read the results of the program’s analysis of this activity. Although the screen is not shown in color here, the LEDs are green in the “Document,” “Activate,” and “Lock” columns, which tells you that this view is fully compatible with BC Sets — in other words, you can create (“Document”) a BC Set for this customizing object, “Activate” it later on, and set the attribute “fix” (i.e., “Lock” fields in this object so they are unchangeable after activation). The “Object type”

Figure 2 The Start Screen of Rupp’s Program



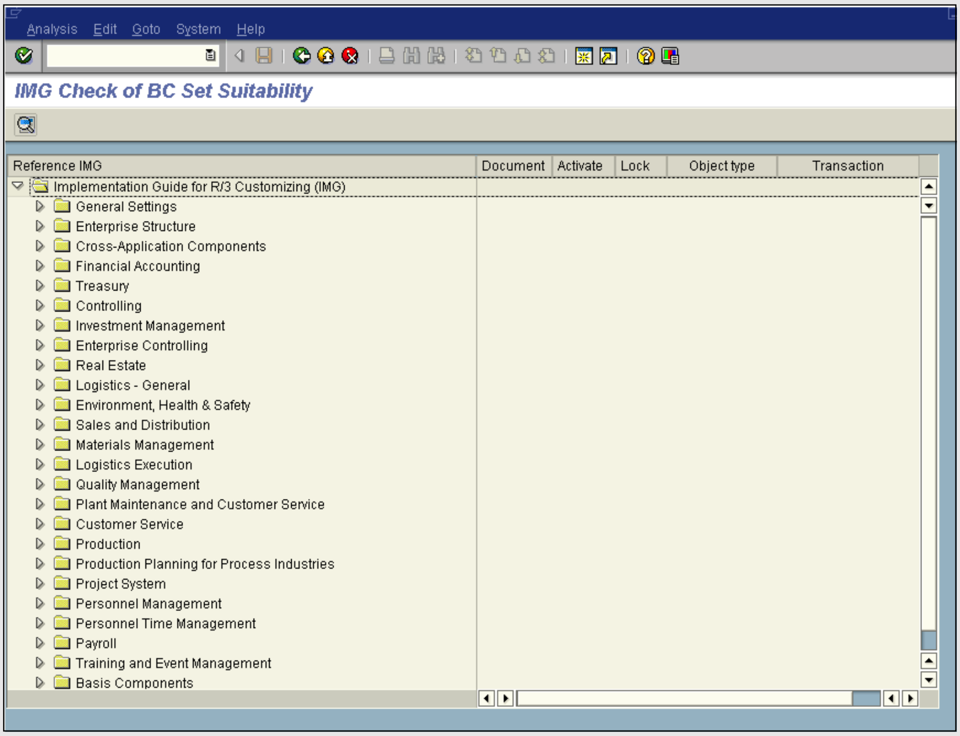
IMG Check of BC Set-Compatibility

Last Check On 25.07.2001

● Display Last Check Result
○ New Check Without Display

[13] Although DSM and BOC had support packages 20 and 19, respectively, they were not aware that this report existed until we introduced it to them.

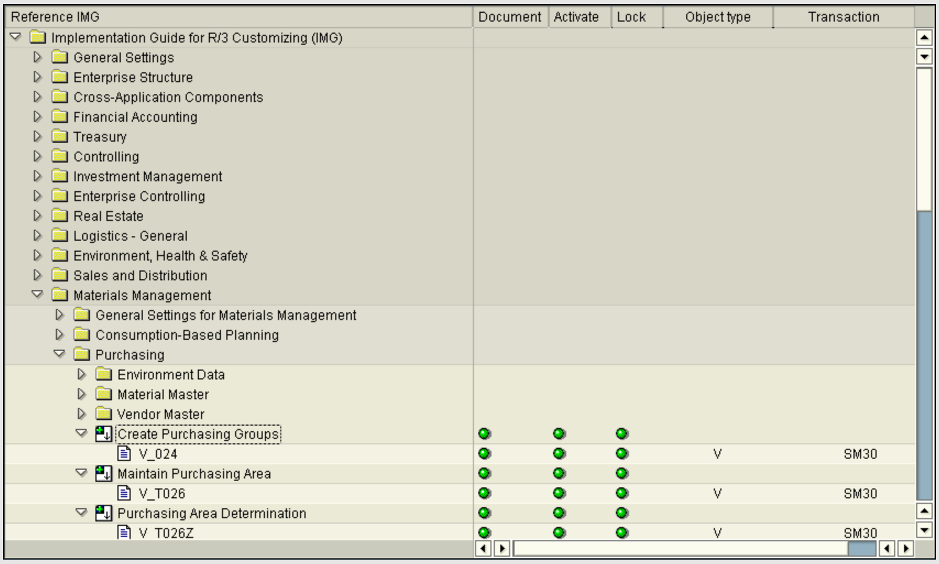
Figure 3 Reference IMG Structure



IMG Check of BC Set Suitability

Reference IMG	Document	Activate	Lock	Object type	Transaction
Implementation Guide for R/3 Customizing (IMG)					
General Settings					
Enterprise Structure					
Cross-Application Components					
Financial Accounting					
Treasury					
Controlling					
Investment Management					
Enterprise Controlling					
Real Estate					
Logistics - General					
Environment, Health & Safety					
Sales and Distribution					
Materials Management					
Logistics Execution					
Quality Management					
Plant Maintenance and Customer Service					
Customer Service					
Production					
Production Planning for Process Industries					
Project System					
Personnel Management					
Personnel Time Management					
Payroll					
Training and Event Management					
Basis Components					

Figure 4 IMG Structure Expanded to Show BC Set Scenarios and Additional Information



Reference IMG	Document	Activate	Lock	Object type	Transaction
Implementation Guide for R/3 Customizing (IMG)					
General Settings					
Enterprise Structure					
Cross-Application Components					
Financial Accounting					
Treasury					
Controlling					
Investment Management					
Enterprise Controlling					
Real Estate					
Logistics - General					
Environment, Health & Safety					
Sales and Distribution					
Materials Management					
General Settings for Materials Management					
Consumption-Based Planning					
Purchasing					
Environment Data					
Material Master					
Vendor Master					
Create Purchasing Groups	●	●	●		
V_024	●	●	●	V	SM30
Maintain Purchasing Area	●	●	●		
V_T026	●	●	●	V	SM30
Purchasing Area Determination	●	●	●		
V_T026Z	●	●	●	V	SM30

and "Transaction" columns provide additional information: the object type is "V" (a view) and the transaction this view was created with is "SM30" (the SAP standard transaction).[14]

If there is a trigger in the activity, the report can tell you that. Highlight a line and choose the "Details" icon (🔍) in the application toolbar. A pop-up box will appear in which you can see whether before-export or after-import methods exist for this object. These methods are what I call "triggers." The behavior of the trigger determines what you can do with BC Sets in this activity.

✓ *Recommendation*

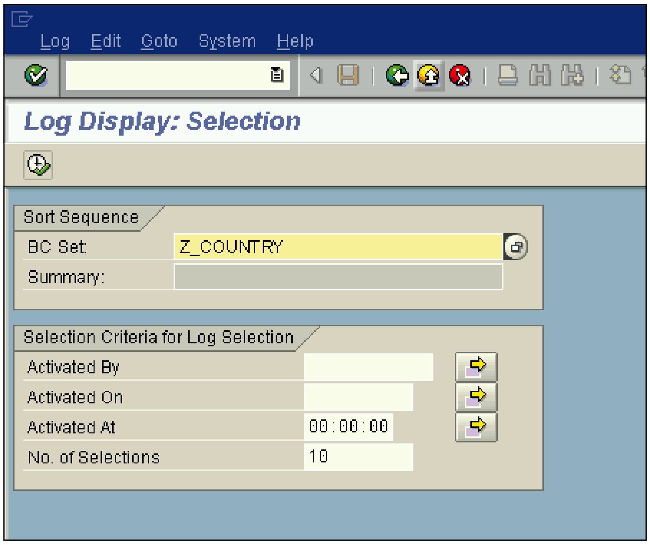
I recommend that you still test the activities that you would like to use. Why? Because the data evaluated by the report is manually entered data, and this means that for a few activities, the result of Rupp's Program and a test may be different. Some of these activities might provide more BC Set functionality than the report indicates. Nevertheless, we are confident that for the majority of IMG activities the report provides the correct results.

Note, however, that you do not need to test objects of type "V" that use transaction SM30, and that show all-green LEDs.

The BC Set Activation Log

When DSM and BOC were carrying out their projects, the log of BC Set activations was not adequate for their needs — it was not persistent and did not contain all the information they required. We subsequently added the first complete log functionality in Basis support package 24 for 4.6C (SAPKB46C24).[15] A refined version of that functionality is available with support package 32 (SAPKB46C32), and that is the version that I will demonstrate for you now:

Figure 5 Select a BC Set for Display



Log Display: Selection

Sort Sequence

BC Set: Z_COUNTRY

Summary:

Selection Criteria for Log Selection

Activated By

Activated On

Activated At 00:00:00

No. of Selections 10

1. To access the log, go to the activation log transaction (SCPR20PR). You will see a screen like the one shown in **Figure 5**. This is the entry screen of the activation log. On this screen, you can select the BC Set for which you want to see an activation history. Here, I have selected the BC Set "Z_COUNTRY" because I would like to find out when this BC Set was activated.

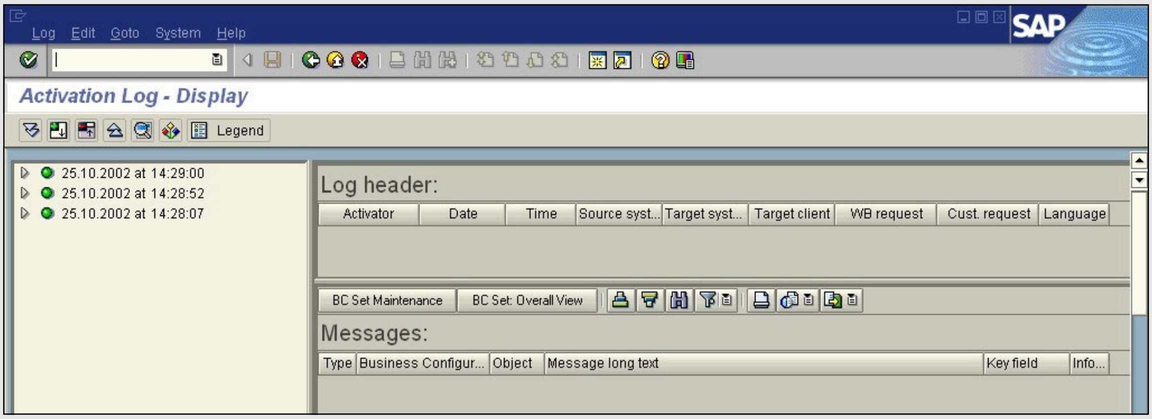
2. To display the activation history of this BC Set, click on the 🕒 icon, which displays the screen shown in **Figure 6**.

   Looking at Figure 6, you see on the left side of the screen a chronological list (in reverse order) of all activations. In this example, the BC Set "Z_COUNTRY" was activated three times on October 25, 2002 — the first activation was at 14:28:07, the second at 14:28:52, and the third at 14:29:00.

[14] Object types that can be used with BC Sets also include "C" (cluster) and "S" (tables). Individually developed objects (type "T") and logical transport objects (type "L") require closer analysis and, generally, testing.

[15] This underlines the importance of having the current support packages when using BC Sets. SAP plans to bring missing functionality to earlier releases, which means that you get the most recent technology for free with support packages.

Figure 6 *Activation History Display*



- 25.10.2002 at 14:29:00
- 25.10.2002 at 14:28:52
- 25.10.2002 at 14:28:07

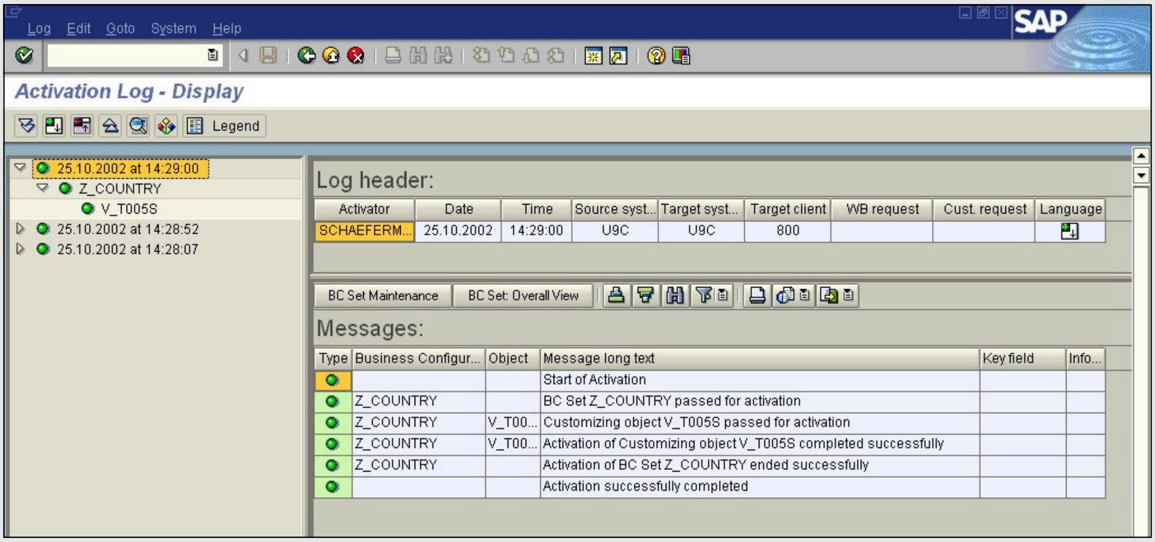
Log header:

Activator	Date	Time	Source syst...	Target syst...	Target client	WB request	Cust. request	Language

Messages:

Type	Business Configur...	Object	Message long text	Key field	Info...

Figure 7 *The Activation Log*



- 25.10.2002 at 14:29:00
  - Z_COUNTRY
    - V_T005S
- 25.10.2002 at 14:28:52
- 25.10.2002 at 14:28:07

Log header:

Activator	Date	Time	Source syst...	Target syst...	Target client	WB request	Cust. request	Language
SCHAEFERM...	25.10.2002	14:29:00	U9C	U9C	800			

Messages:

Type	Business Configur...	Object	Message long text	Key field	Info...
●			Start of Activation		
●	Z_COUNTRY		BC Set Z_COUNTRY passed for activation		
●	Z_COUNTRY	V_T00...	Customizing object V_T005S passed for activation		
●	Z_COUNTRY	V_T00...	Activation of Customizing object V_T005S completed successfully		
●	Z_COUNTRY		Activation of BC Set Z_COUNTRY ended successfully		
●			Activation successfully completed		

3. To find out what happened during a particular activation, open the tree next to the date/time of the activation. In this example, I open the tree for the latest activation (see **Figure 7**). All BC Sets contained in the BC Set are listed in the tree. In this example, the BC Set “Z_COUNTRY” does not contain other BC Sets, since it is not a hierarchical BC Set. Therefore it is the only BC Set listed. On the next level down, you can see which views/tables are contained in the BC Set.

   In this example, BC Set “Z_COUNTRY” only consists of one view (“V_T005S”).

4. To display the activation history, double-click on the activation (in the example, “25.10.2002 at 14:29:00”). The activation information is listed on the right side of the screen in two different frames: “Log header” and “Messages.”

   Under “Log header,” you can find the following:

- **Activator/Date/Time:** These fields show you who activated this BC Set and when.

- **Source system:** This field shows whether the activation was started from the logon system or from a different system (like the Solution Manager system[16]). In the example, the value "U9C" is local (i.e., the activation was started in this system).

- **Target system:** This field shows in which system the activation was started. In this example, the value is "U9C" again, because I did this demo while logged on to the local U9C system.

- **Target client:** This field shows in which client the activation was started. In the example, it is the activation client "800."

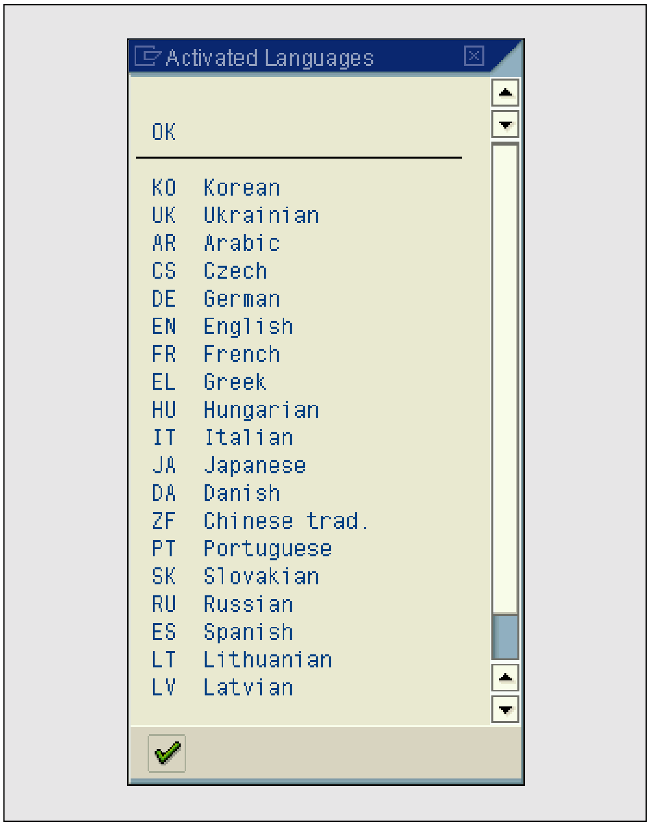
- **WB request/Cust. request:** If the client in which the BC Set is activated is set to "recording on," then requests are being created. The names of these requests are stored in the "Workbench request" and "Customizing request" fields. In the example, because I performed this demonstration in an internal system, the client does not automatically record customizing changes in transport requests. Both fields are therefore empty. In a real-world customer project, the customizing client is always set to "recording on," so the appropriate field is filled. This information is vital because these requests have to be channeled through the QA system to the PRD system. This information is also vital if errors are reported in QA or PRD, and you want to trace the transport back to the BC Sets that activated this customizing setting.

- **Language:** This field is only relevant for language-dependent entries, such as descriptions of text fields. If you have such entries in your BC Set, you need to know which translations of the entry are included in the BC Set. To find out, click on the icon, which launches the pop-up shown in **Figure 8**. This dialog lists all languages into which language-dependent entries are translated. Keep in mind that the translation of a language-dependent entry must exist in the system where the BC Set was created prior to the BC Set creation.

Under "Messages" in Figure 7, you see the messages that were created during the individual steps of the activation process. The following information is available:

- **Type:** This column displays the result of an activation step. The result can be green (no error), yellow (warning), or red (problem). In the example, none of the activation steps encountered problems, so they are all green.

Figure 8 Languages into Which a Language-Dependent Entry Can Be Translated



KO	Korean
UK	Ukrainian
AR	Arabic
CS	Czech
DE	German
EN	English
FR	French
EL	Greek
HU	Hungarian
IT	Italian
JA	Japanese
DA	Danish
ZF	Chinese trad.
PT	Portuguese
SK	Slovakian
RU	Russian
ES	Spanish
LT	Lithuanian
LV	Latvian

[16] For more on the Solution Manager, see my article "Are You Certain You've Got Consistent Customizing Settings Across Your SAP Landscape?" (March/April 2002).

- **Business Configuration Set:** This column lists the names of the BC Sets that raised the message. If you activated a hierarchical BC Set, it is important to know which of the embedded BC Sets caused an activation error.

- **Object:** This column shows the name of the object involved in the activation step. If you have problems with a BC Set, it is useful to find out which object(s) inside the BC Set caused problems.

- **Message long text:** This column displays the message text.

- **Key field:** This column shows the key field(s) that caused an error. As the example in Figure 7 shows, this column is not always filled when the BC Set causes an activation problem. If it is filled, it will contain the keys for the table lines that could not be written to the customizing table in the system.

- **Information:** This column provides access to an explanation of the error that occurred.

DSM and BOC were delighted when we reviewed the activation log with them, and they are currently using the enhanced log in their projects.

Final Lessons Learned

☑ **Knowledge transfer is critical.** As BC Sets were still somewhat new and not quite mainstream, there was not a great deal of knowledge and expertise available to the DSM and BOC teams during their projects. The consultant teams and the internal teams, for the most part, had no previous experience with BC Sets. BC Sets also had to be "sold" internally to IT executives. So knowledge transfer became a key factor for the success of the DSM and BOC rollout projects — not just in terms of preparing customizers and local teams, but at every level of the project.

☑ **Be prepared to move to the highest Basis support package level while using BC Sets.** We continue to expand functionality for earlier releases.

☑ **Plan your BC Sets carefully.** Create small BC Sets first, then use hierarchical BC Sets to combine them into bigger BC Sets.

☑ **Be restrictive.** Put only the minimal amount of data that belongs to your template in BC Sets.

Conclusion

In February 2002, DSM and BOC were among the companies that came to Walldorf for the first BC Set expert user exchange. When I asked the team leaders whether they would use BC Sets again, both said, "Overall, yes." They appreciate the advantages of the BC Set technology — the ability to track customizing settings, protect the kernel with locked settings, transfer or reuse existing knowledge or system parameters, and build into their templates the flexibility to allow subsidiaries to localize certain settings. As one leader put it, "[BC Sets] is a great technology, but it has some sharp edges if you remain on 'old' support packages." Both teams agreed that the new functionality provided by the BC Set development team at SAP is on the right track.

If you've been sitting on the sidelines with regard to BC Sets up until now, watching others implement this technology to see whether it is worth adopting for your own global rollouts, it just might be time to consider jumping in. The technology is maturing, the tools are better than ever, and the savings in time and costs are there for the taking. In R/3 Enterprise, the next release of R/3 4.6, the use of BC Sets in the IMG will be improved significantly. More IMG activities will allow the creation and activation of BC Sets, and locking will be available in more activities than ever.

Additional Information Resources

SAP Service Marketplace

The BC Sets home page in the Service Marketplace (**http://service.sap.com/bcsets**) is a comprehensive resource.

We also produced two sessions on BC Sets and one on the Customizing Cross-System Viewer (CCSV), all of which can be downloaded from the Customizing and Testing Tools home page in the Service Marketplace (**http://service.sap.com/customizing**). Select *Media Library → Presentations*.

BOC (www.boc.com)

For more information on how BOC has used BC Set technology in their global rollout project, or for any questions about BOC, please contact Brian Egles, manager of BOC's global template team. Brian can be reached at Brian.Egles@uk.gases.boc.com.

DSM (www.dsm.com)

For more information on how DSM has used the BC Set technology in their global rollout project, or for any questions about DSM, please contact Peter Graus, team lead of the ICT Infrastructure and Integration Management of DSM's Corporate ICT department. Peter can be reached at Peter.Graus@dsm.com.

Acknowledgements

I would like to thank Brian Egles of BOC and Peter Graus of DSM, whose invaluable help and insights made this article possible. I would also like to thank Pierre De Winter of Deloitte Consulting, who recognized the need for BC Set technology in global template rollouts and became an early adopter of this technology. Your help is very much appreciated.

Dr. Matthias Melich studied English Literature at the University of Rochester (New York), and Mathematics and English Philology at the University of Cologne. He also received his Ph.D. in Computer-Based Language Learning at the University of Cologne in 1993. He joined SAP in 1995 as a member of the Common Objects and Modeling team, and a year later joined the Archiving Solutions team. Since 1998, Matthias has been responsible for Product Management of Customizing Tools. In January 2001, he became a member of the Solution Management Technology department, which is responsible for the concepts and development of the SAP Solution Manager. Matthias is Manager of the Technical Product Integration team, which focuses on knowledge management, product planning, and development of implementation standards and guidelines. He is also the author of "Computer-Based Training for R/3 Archiving" and the SAP DocuSet "Data Modeling with the Data Modeler." He can be reached at matthias.melich@sap.com.