

# Manually registering a non-SAP system in SMSY

by Doreen Baseler

As you learned in the article “Ensure effective system management by configuring the right infrastructure for SAP Solution Manager using central system landscape maintenance” (*SAP Professional Journal*, September/October 2005), in order to manage the SAP systems in your landscape using SAP Solution Manager, you need to register and classify them in transaction SMSY (Solution Manager System Landscape). But what about the non-SAP systems in your landscape? While you cannot perform the kind of in-depth maintenance on your non-SAP systems that you can on your SAP systems, such as automatic system data retrieval, you can manually register them in SMSY for a complete picture of your system landscape and make them available for use in SAP Solution Manager projects and solutions. Here we’ll take a brief look at how to do this.

## Note!

Keep in mind that the definition of non-SAP products and the use of such systems in your landscape is entirely customer-specific. You will need to tailor the information provided in the following sections to your own unique environment and purposes.

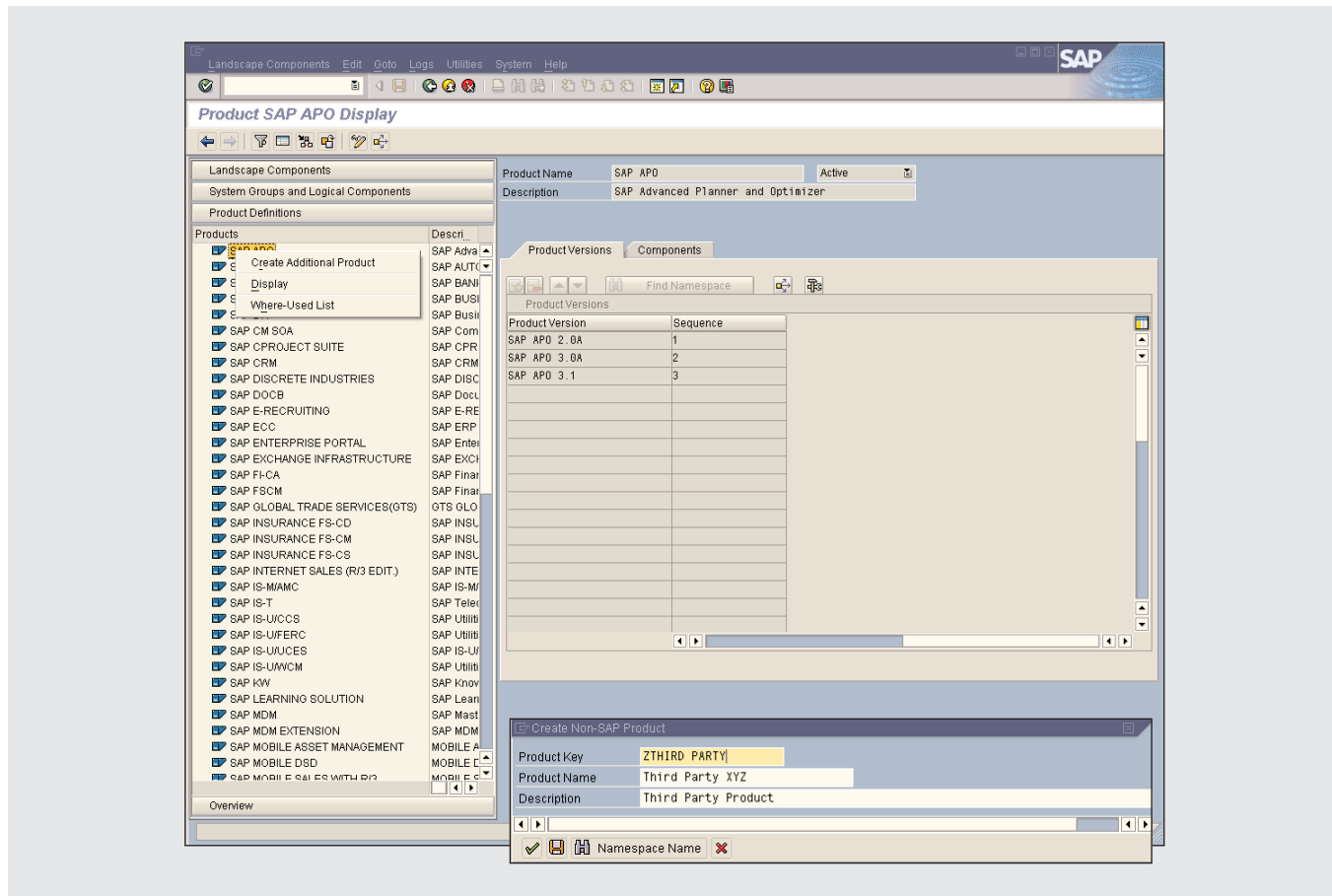
## Step 1: Create a new product definition for the non-SAP system

The first step is to make sure that a product definition has been created for the non-SAP system. Definitions for SAP products are automatically shipped with SAP Solution Manager<sup>1</sup> and displayed in the Product

<sup>1</sup> SAP product definition updates are shipped with SAP Solution Manager Tools (ST) and content add-on (ST-ICO) support packages or corresponding support package stacks.

Definitions view in transaction SMSY.<sup>1</sup> Product definitions for non-SAP products, on the other hand, must be defined manually:

1. In transaction SMSY, open the Product Definitions view in the left pane, right-click on any item in the Products tree, and choose Create Additional Product from the context menu (see **Figure 1** on the next page).
2. In the dialog box that appears, also shown in Figure 1, enter a product key that uniquely identifies the product (e.g., ZTHIRD PARTY), a product name (e.g., Third Party XYZ), and a description (e.g., Third Party Product). Save your entries.
3. On the Product Versions tab (see **Figure 2** on page 3), the system automatically displays the following data:
  - A *product version key*, which uniquely identifies this particular version of the product. An initial value for the product version key is proposed by the system. By default, the value of the product version key is the value defined for the product key when you created the external product (ZTHIRD PARTY in the example). You can change this value if you want to reflect the true software numbering or versioning scheme of the respective product. Since the product version key serves as a unique identifier, it cannot be changed once it is saved, however.
  - A *product version*, which is the descriptive representation of the specified product version key. The product version key serves a technical purpose, whereas the product version name representing the key is offered in the possible entries help when you create a system based on that product. An initial value, which is generated automatically and always starts with “01,” is proposed



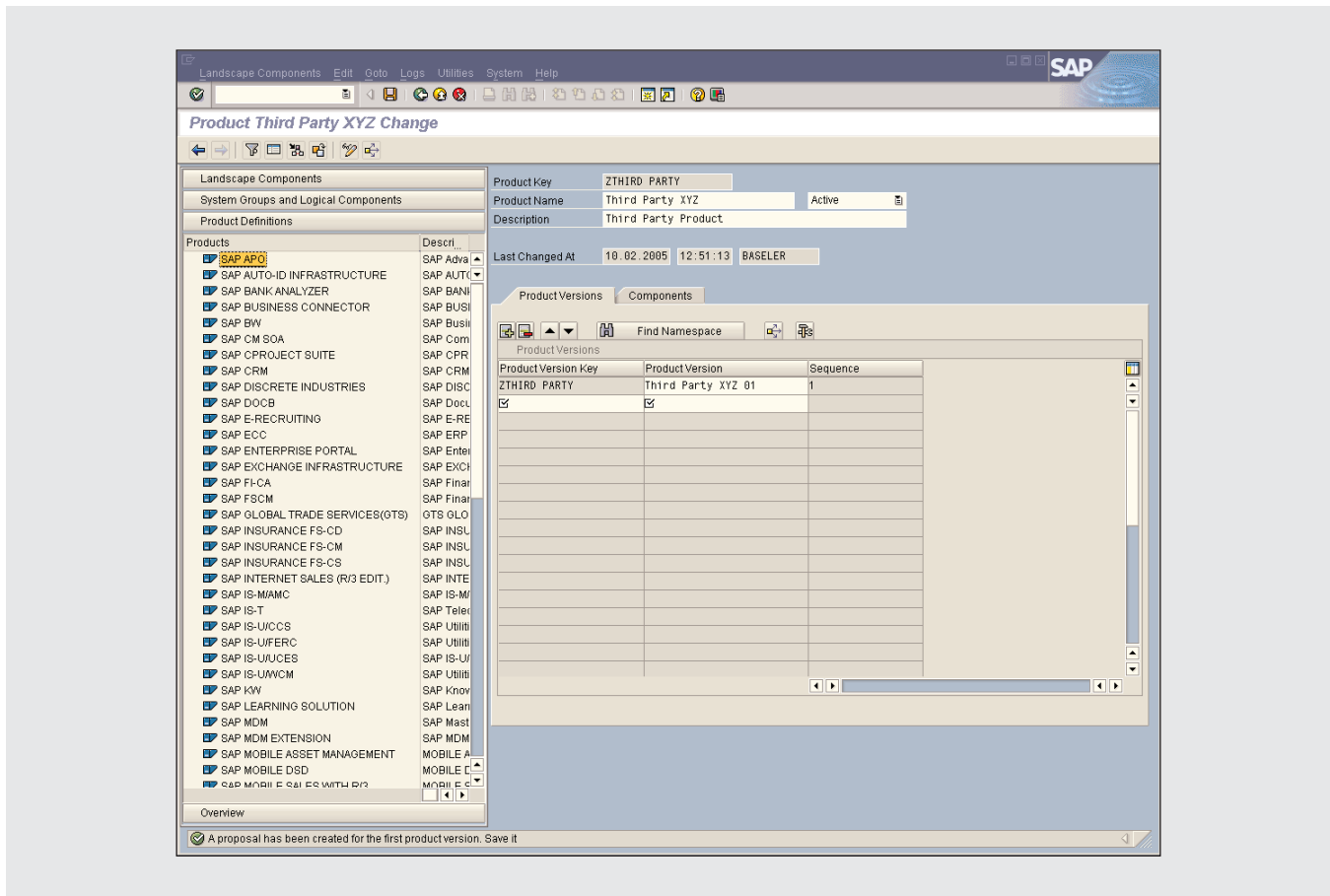
**Figure 1** Creating an external product

by the system, but you can change or enhance this value to be more descriptive. In the example, the suggested product version is Third Party XYZ 01. If you are not satisfied with the proposed product version, you can change it either immediately or after saving. If you are additionally dissatisfied with the proposed product version key, you must delete the complete line and add a new entry. Remember that after saving your entries, the product version key is fixed in order to guarantee the uniqueness of the corresponding key.

- A *sequence*, which defines the predecessor-successor relationship for the product versions. As you can see in Figure 2, the number “1” has been assigned as the sequence value for the example product version, since it is the first defined product version. The sequence

value is proposed by the system and cannot be changed manually. You can, however, change the sequence for product versions by using the Move Current Row Up and Move Current Row Down buttons (▲▼).

4. Save the settings. Additional product versions can be added directly on the Product Versions tab by submitting new keys and descriptions for each product version. Again, note that after saving you can make changes only to the product version value. If you are not satisfied with the product version key, you must delete the complete line and add a new entry.
5. On the Components tab, specify the main instances of the new product (see **Figure 3** on page 4). Unlike the process for SAP products, this is a manual step.



**Figure 2** Defining the product versions

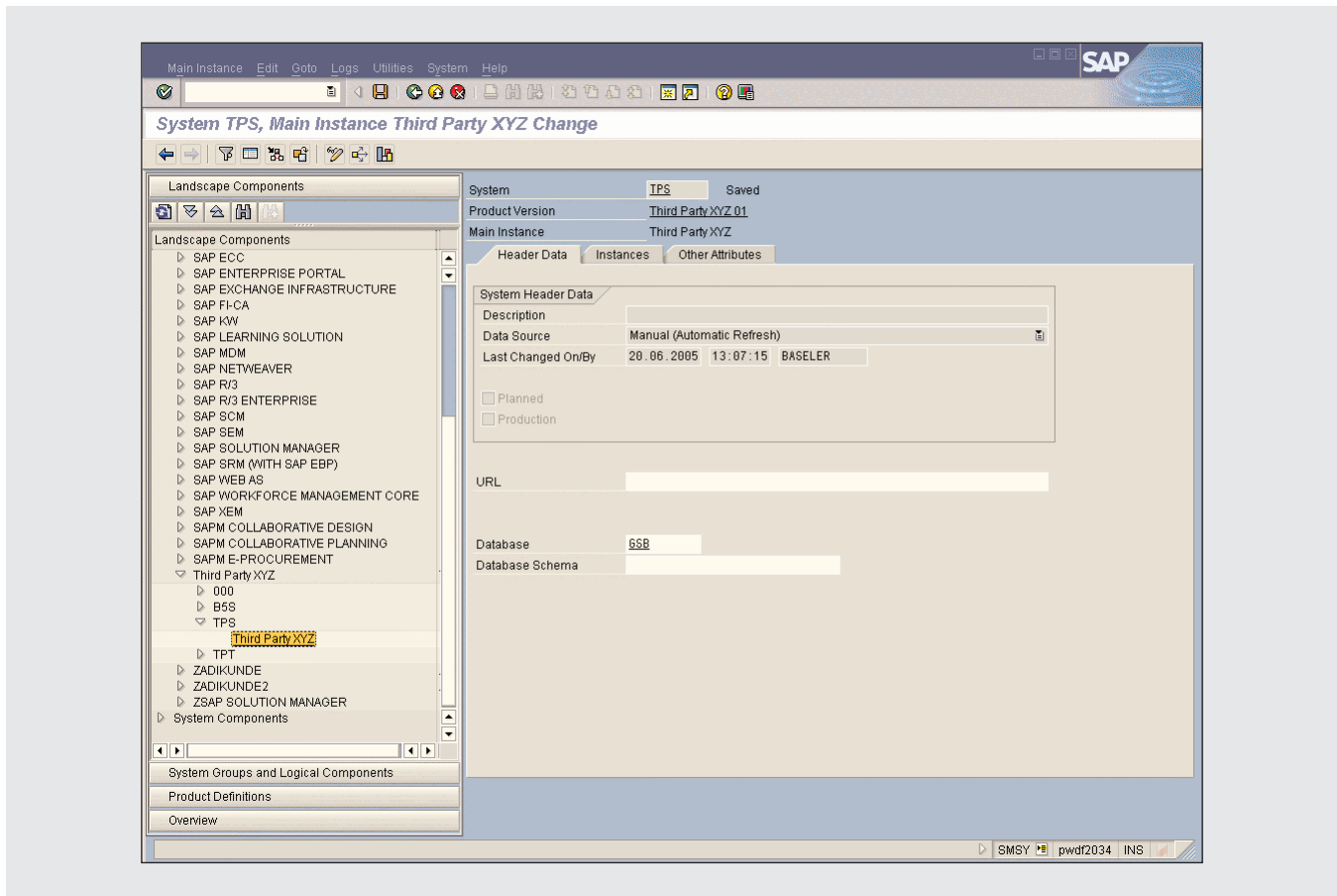
### **Note!**

Selecting the With Database option on the Components tab enables you to assign a system's database to the external product along with the system. This means that SMSY allows you to maintain a complete picture of your system landscape, not only including products and systems but also the databases that store the data used by the systems. For SAP systems, this information can be maintained directly on the system level. As with SAP systems, you can report on this information anytime via SAP Solution Manager System Landscape Reporting.

Here the software packages of the product are portrayed as SAP main instances that are relevant for implementation and monitoring activities. In the example, Third Party XYZ is the suggested value, which was derived from the product itself. If the suggested value does not meet your demands, you can

simply delete the proposed main instance and create one yourself. It is essential that you define at least one main instance, because when later creating a system based on that product, you need to flag at least one main instance as relevant for the system, as described for SAP systems in the article.





**Figure 4** Third-party system and relevant main instance in SMSY

7. Save your entries. The product is added to the product list, as shown in Figure 3. You can easily distinguish between SAP products (📁) and non-SAP (external) products (👤) via the corresponding icons.

## Step 2: Maintain the system information for the non-SAP system based on the non-SAP product

The procedure for registering a non-SAP system in SMSY is the same as for SAP systems, but with a reduced scope of settings:

1. In transaction SMSY, open the Landscape

Components view and create a “shell” for the non-SAP system as described for SAP systems in the article. For the example, I have created a third-party system called TPS based on the Third Party XYZ product defined in Step 1. After saving, the system is automatically listed in the SMSY Landscape Components tree.

2. Assign the relevant main instance on the Selection of Main Instances tab, following the same procedure outlined for SAP systems in the article. After saving, the selected main instance (Third Party XYZ in the example) is added below the system in the SMSY Landscape Components tree (see **Figure 4**).
3. As with SAP systems, you also have the option to provide additional details on your non-SAP system.

Figure 4 shows the tabs containing the available details — Header Data, Instance, and Other Attributes. The Clients and Software Components tabs available for SAP system main instances are not available for non-SAP main instances, since these tabs deal exclusively with SAP-specific information. The Header Data tab for non-SAP system main instances also does not include SAP-specific information like system number, message server, installation number, and transport domain. The information you choose to maintain is entirely up

to you. For the example, you can see that I entered instance database information, which allows me to report on the database using system landscape reporting.

We have now completed our brief tour through defining non-SAP systems and products in SMSY. Even though I did not cover all aspects of it in detail, you now have a foundational knowledge of the tasks involved that you can use to shape the approach you'll take with your own project.