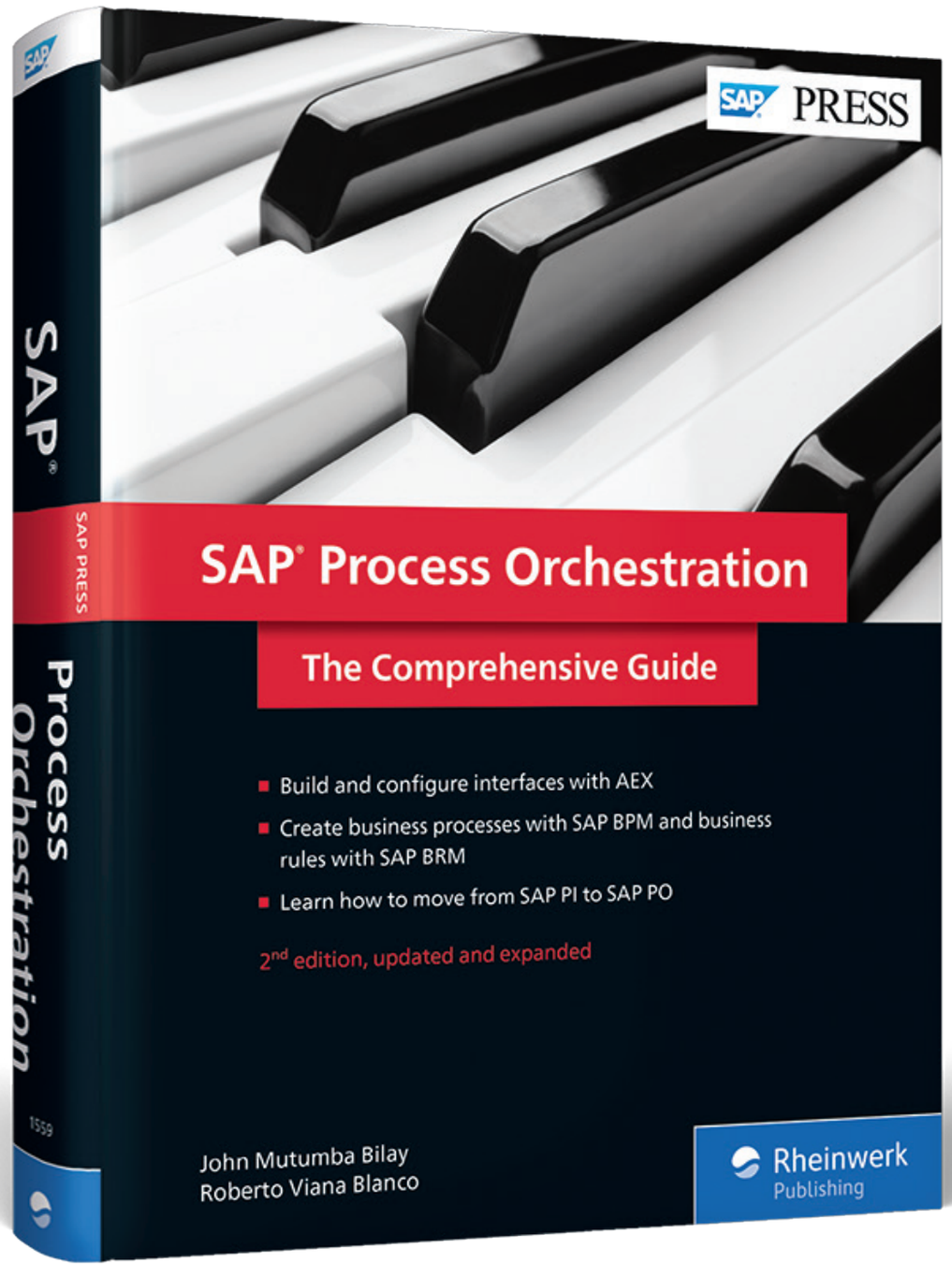


Reading Sample

This sample chapter explains the role of the System Landscape Directory (SLD) in the context of SAP integration. It shows you how to register systems to the SLD, what administration of the SLD looks like, and provides some helpful tips and tricks for working with the SLD. It also contains a practical exercise for you to test what you've learned.

"Configuring the System Landscape Directory"

Contents

Index

The Authors

John Bilay, Roberto Viana Blanco

SAP Process Orchestration: The Comprehensive Guide

908 Pages, 2017, $79.95
ISBN 978-1-4932-1559-1

www.sap-press.com/4431

Chapter 3

Configuring the System Landscape Directory

Finally we shall place the Sun himself at the center of the Universe.

—Nicolaus Copernicus

The System Landscape Directory (SLD) plays a critical role in an SAP system landscape. As a central repository, it manages information about all installable and installed elements of your system landscape in your organization. The repository includes a list of SAP and non-SAP systems from a technical and business perspective, and the information available is used by other SAP applications, such as SAP Process Orchestration (SAP PO), SAP Solution Manager, SAP NetWeaver Administrator, and the SAP NetWeaver development infrastructure.

The information contained in the SLD also includes which products and software component versions (SWCVs) are installed in a particular application or system. You can make systems that participate in the landscape automatically register themselves and update the SLD with recent information as they become available. As time goes by, the initially installed SLD content becomes outdated and will need to be updated with a more recent version. The most up-to-date SLD content can be found on the SAP Service Marketplace.

You can access the SLD via *http://<hostname>:<port>/sld*, where *<hostname>* is the hostname of the SAP NetWeaver system and *<port>* is the port number of the SAP NetWeaver system. The SLD is a Java-based component that can be installed on an SAP NetWeaver Application Server for Java (SAP NetWeaver AS Java). The SLD is included in the `SAP_JTECHT` software component (SC) delivered by SAP.

A newly installed SLD comes prepopulated by default with some initial data provided by SAP, including a catalog of all possible installable SAP products, SCs, and details about support packages. You can use this default data and these SCs as dependencies for your own custom content.

In this chapter, we'll explain the SLD's roles within the context of SAP integration. We'll discuss the SLD components and explain their functionalities. This chapter also includes an exercise, which serves as an implementation example.

3.1 System Landscape Directory Components and Features

To fulfill its role of central information repository for the entire system landscape, the SLD facilitates easy access to the information of different systems and software. In the **System Landscape Directory** landing page (see Figure 3.1), the information is categorized in the following sections or categories:

- **Landscape**
- **Software Catalog**
- **Development**

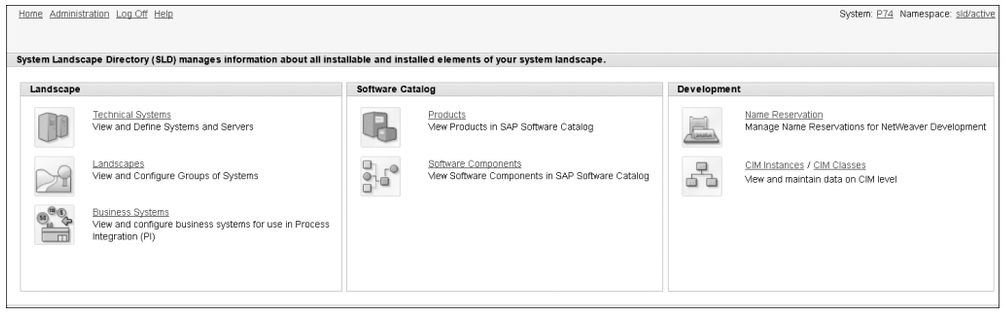


Figure 3.1 System Landscape Directory Landing Page

In the next sections, we'll explore each of these categories in more detail.

3.1.1 Landscape

The **Landscape** category includes functionalities covering the maintenance of technical systems, landscapes, and business systems.

Technical Systems

Technical systems are SAP and non-SAP backend applications in your landscape. Any SAP components (SAP ERP Human Capital Management [SAP ERP HCM], SAP Customer Relationship Management [SAP CRM], etc.), SAP NetWeaver applications,

third-party applications, or legacy systems in your landscape can be considered technical systems to be maintained in your SLD. If an application registers itself with the SLD (as discussed in Section 3.2), then a technical system will be created in the SLD to represent it.

Assume that an SAP CRM system has a system ID of ECD. In that case, a technical system with the name ECD will be created in the SLD. As demonstrated in Figure 3.2, a lot of information about the ECD system is maintained in the SLD. Some of this information includes the following:

- System name
- Hostname of the server
- Version number of the system
- Type of system
- When the data was last updated
- Details about the database (see the **Database** tab)
- List of existing clients (see the **Clients** tab)
- Details of the message server of the application

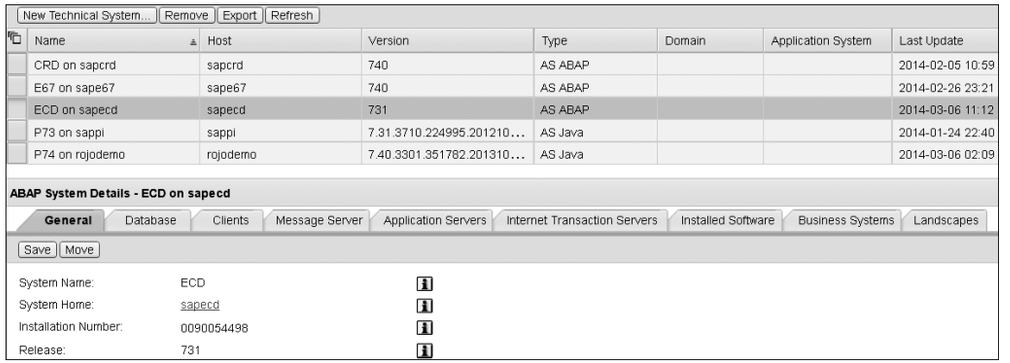


Figure 3.2 A Technical System in the SLD and Various Tabs Containing Its Data

If a system wasn't automatically registered in the SLD (especially in the case of a third-party system), then you can manually create a new technical system. To do so, click the **New Technical System** button (as shown in Figure 3.2) and follow the wizard. You'll be required to specify the type of technical system to be created. The possible choices include the following:

- **SAP NetWeaver AS ABAP**
  Systems with an SAP NetWeaver AS ABAP. Most SAP backend systems will fit into this category.
- **SAP NetWeaver AS Java**
  Systems with SAP NetWeaver AS Java. Such a system can consist of one or multiple instances.
- **Standalone**
  Standalone Java applications.
- **Third-party**
  Systems containing third-party (non-SAP) applications and products.

Landscapes

Landscapes provide the facility to group systems in a logical manner. A landscape can be seen as a development environment that includes a number of servers and applications intended for a specific stage in a release process.

For each landscape, you can create a group in the SLD and include all applications and systems belonging to that stage. To create a new landscape for the development stage, follow these steps:

1. Choose **Landscape** from the **System Landscape Directory** landing page.
2. Click on **New Landscape** to define a new landscape.
3. Add all the relevant systems belonging to this stage (see Figure 3.3).

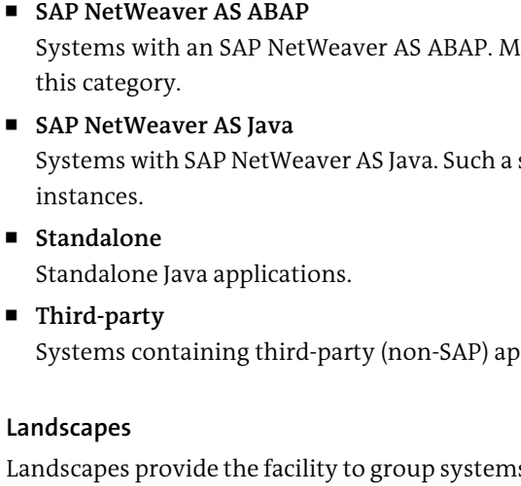


Figure 3.3 Adding Different Systems to a Landscape

You can add systems to a particular landscape by performing the following steps:

1. Select the desired landscape.
2. Click the **Systems** tab.
3. Click the **Add System** button.
4. Choose the type of system from the **Type** dropdown.
5. Select the desired system, and click **OK**.

Business Systems

A business system is a logical name given to a particular system or application. In general, the business system name is unique in the landscape and is used in an SAP PO scenario to represent the system or application as a sender or receiver. There is a direct relationship between a business system and a technical system. Every technical system can have at least one business system. For an SAP ABAP backend system, every client can have its own business system. This is only relevant for SAP NetWeaver AS ABAP systems because SAP NetWeaver AS Java-based systems aren't partitioned in clients.

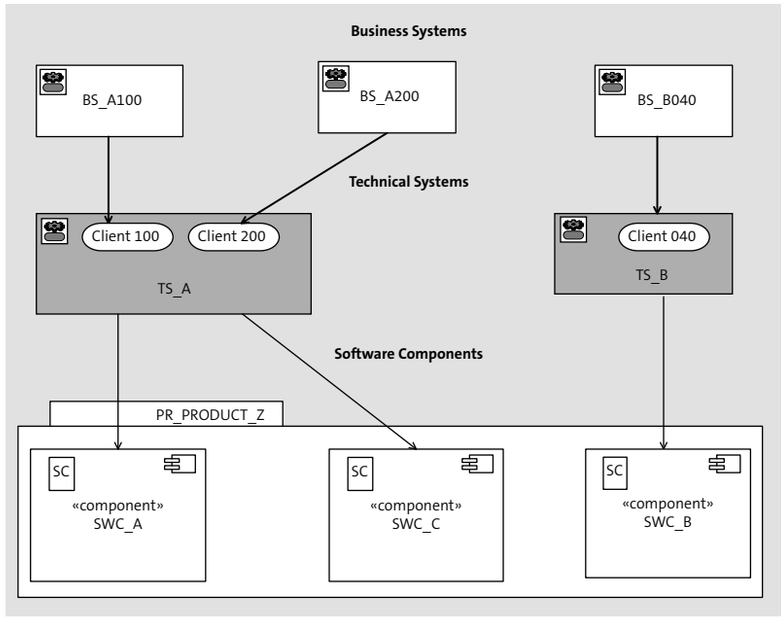


Figure 3.4 Relationship among Business Systems, Technical Systems, and Software Components

Figure 3.4 depicts a situation with a technical system called TS_A that has two clients (100 and 200). Each client is linked to a different business system—namely, BS_A and BS_B.

> **Note**
>
> The backend systems (ABAP) regularly retrieve their business system names from the SLD. If the SLD becomes unavailable, then the SAP backend system won't be able to update its current business system name and might be using an outdated business system name that it keeps in its cache.

You can perform all management tasks from the SLD: create, update, delete, and search or filter on business systems. Compared to technical systems (which are created automatically when the application systems register themselves to the SLD), the business systems need to be manually created. To create a business system, follow these steps:

1. From the **System Landscape Directory** landing page, choose **Business Systems • New Business System** to start the wizard.
2. Select the type of technical system that the new business system will be linked with.
3. In the next step of the wizard, you'll be asked to choose the related technical system and client (for the ABAP system).
4. Give your business system a logical name.
5. Choose the products and SCs that are installed in your business systems.
6. Select the role of your business system. The following options are available:
   - **Application System**
     This option will be used most of the time and is relevant whenever the business system to be added represents a backend application or system, in which case, it's required to specify its *related integration server*. The related integration server represents the SAP Process Integration (SAP PI)/SAP PO system that will act as the enterprise service bus (ESB) for the business system in question (see Figure 3.5).
   - **Integration Server**
     This is relevant in cases in which the system in question is middleware: an SAP PI or SAP PO system.

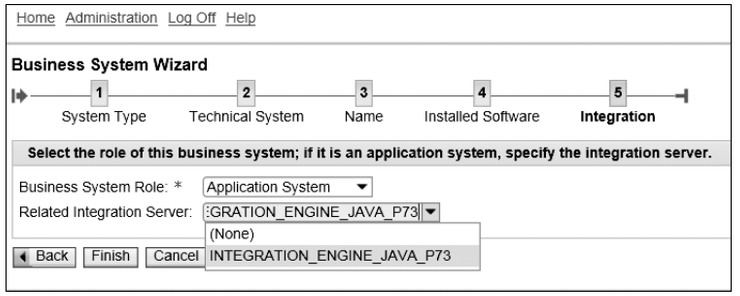


Figure 3.5 Specifying a Business System's Role While Creating a New Business System

A business system can be described by a number of attributes. After selecting a particular business system from the SLD, a few tabs are presented: **General**, **Integration**, **Transport,** and **Installed Software** (see Figure 3.6).

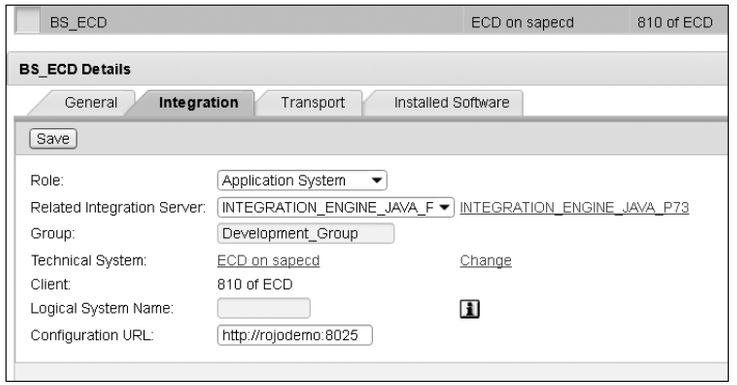


Figure 3.6 Business System: Integration Details

When dealing with business systems, it's very useful to understand the concept of business system groups, which represent a grouping of business systems that belong to the same environment, thus effectively dividing your landscape into different controllable parts. Most organizations have the following four environments in their landscapes:

- Development
- Test
- Acceptance
- Production

In most organizations, systems belonging to the same group will often be named in similar ways. Of course, this depends very much on the organization's naming conventions. For example, all systems of the development environment might have the format `SYSTEMNAME_D`, where `D` represents "development," and all test systems might have the suffix `_T`. In this way, it's easy visually to identify the environment that the system belongs to based on its name.

It's common to have a group of business systems defined in the SLD for each one of the environments or stages of your landscape. SLD groups play an important role in managing transport targets. To manage groups, follow these steps:

1. From the **System Landscape Directory** landing page, select **Business System**.
2. From the **Groups** dropdown, choose **Edit Groups**.
3. Select **New Group**.
4. Give the group a name, and select an integration server responsible for all integrations in this group (this is basically the SAP PO server for this group).

You don't have to add a business system to the group; it's only required to link a group of business systems to an integration server. In addition, all the business systems linked to the same integration server automatically belong to the same group.

3.1.2 Software Catalog

The SLD is also equipped with a repository of SAP products and SCs (see Figure 3.7). This repository contains a catalog of SCs and their dependency relationships with each other. In the software catalog, you can differentiate products and SCs.

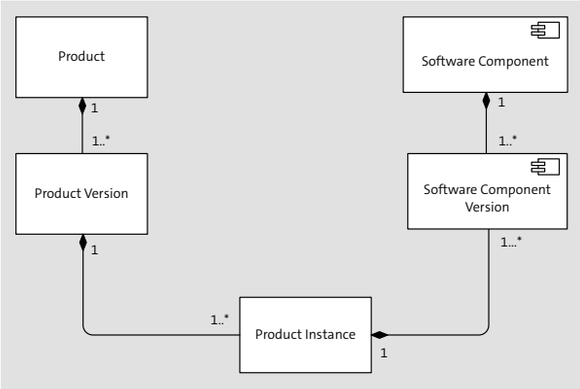


Figure 3.7 Relationships among Products and Software Components

In the next sections, we'll explore products and SCs in more detail.

Product

A *product* is an installable unit of software that includes a logical bundling or grouping of multiple SCs. A product is the highest object in the software catalog hierarchy. A product has a version.

As illustrated in Figure 3.7, the intention is to group SCs in a logical way together into the same product. Therefore, a product called "Banking" assumes a grouping of various SCs related to banking services. To create a new product, follow these steps:

1. From the **System Landscape Directory** landing page, select **Products**.
2. Select **New**, and follow the wizard.
3. You'll be prompted to create a SC as part of this wizard. Be aware that it's also possible to add more SCs later.

Software Component

A *software component* (SC) is a reusable module of a product. It can be upgraded or patched for bugs. A SC is the foundation on top of which an integration developer will develop his interfaces and mappings.

SCs can be dependent on other SCs. The dependency encourages reusability and is also called usage dependency. Usage dependency is used to define a relationship toward other SCs and therefore reuse their functionality.

When setting up dependencies, it's possible to choose from among the following options:

- **InstallationTime**
  With this dependency type, a particular SC needs another SC to be installed. This SC can't work without the dependencies being installed first.
- **BuildTime**
  This dependency type defines the dependencies that are required during compilation of sources and archives. The SC's sources need the resources of the SC that it depends on to be built and compiled.
- **MetaDataRequest**
  This dependency type is used when the metadata of another SC is required during the installation time.

To create a new SC, follow these steps:

1. From the **System Landscape Directory** landing page, select **Software Components**.
2. Select **New**, and follow the wizard.
3. Select the product version and product instance that this new SC should belong to.
4. Specify the SC name and version.
5. Click **Finish**.
6. After successfully creating a SC, you can create dependencies by clicking the **Dependencies** tab.
7. From the **Context** dropdown, select the desired dependency type: **BuildTime**, **InstallationTime**, or **MetaDataRequest** (see Figure 3.8).
8. Select the **Define Prerequisite Software Component Versions** option and the required SC.

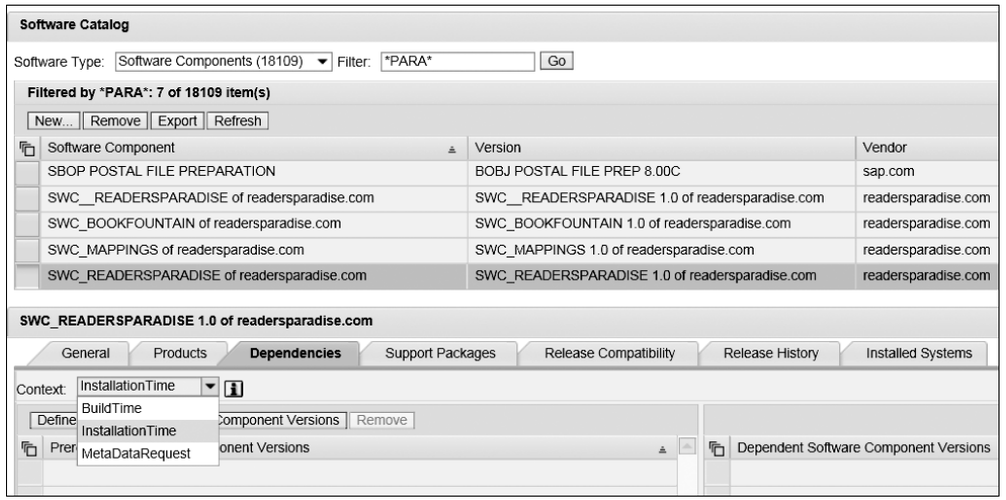


Figure 3.8 Adding Software Component Dependencies in the SLD

To properly manage the SCs in your landscape, you need to have a good approach for how to classify and organize them. This is also referred to as the SLD SC model. The SLD SC model helps you determine how many SCs should be used to represent a particular integration scenario. It's important to think about the component model or strategy to be used and to choose one that fits best in your situation. The SLD SC model will typically be part of your development standard.

In general, three SC model strategies are widely used (see Table 3.1 for more details).

Model Name	Description
One SC model (also referred to as the horizontal approach)	In this approach, a single SC is created for all interfaces of the entire organization. Namespaces are then used to separate the different domains or groups of objects. The obvious big disadvantage of this approach is that there is no reusable unit of software.
Two SC model (also called the vertical approach)	In this approach, two SCs are created: one encapsulating the sender and the other encapsulating the receiver. All objects and message structures belonging to the sender systems are placed in one SC. The same applies to the SC of the receiver system. The question to be asked with this approach is where to place common or shared objects (e.g., mappings) to make the translation between the sender's message structure and the receiver's message structure.
Three SC model	This approach uses three SCs: the first for the sender system, the second for the receiver system, and the third for all shared objects. This means that objects such as mappings should be placed in the third software component. This model represents a clear separation between the different SCs and encourages reusability. This model is the most commonly used.

Table 3.1 Software Component Models

Note

You can also come up with your own SC model and a strategy that will best suit your own needs.

3.1.3 Development

This section of the SLD contains tools designed to facilitate SAP NetWeaver development activities: name reservation and Common Information Model (CIM) instances and classes. We'll explore each of these tools and explain their roles.

Name Reservation

All organizations should have a naming convention. In addition to all the benefits stated in the previous section, a good naming convention allows the developers to create unique development objects and avoid name clashes and confusion between different objects. This is especially relevant when different developers and teams are working on the same project. In a typical project, not everyone is allowed to define the name of the development component (DC) as they wish. These names are predefined by an administrator, and the developers only need to use them. This approach introduces a high level of control and avoids inconsistent definition of DC names.

Name reservation (or the name server) enables the Java development team to define and reserve some names globally throughout the entire landscape. This guarantees the uniqueness of the reserved name. It's also possible to use the namespace concept to reserve a unique name in the name server.

A namespace always starts with a namespace prefix. Namespace prefixes defined in a name reservation can later be used in the SAP NetWeaver development infrastructure. You should use your Internet domain names in the namespace definitions to make sure they are globally unique.

It's important to note that the SLD isn't automatically activated as the name server for your SAP NetWeaver development infrastructure developments. You can activate the name server role in the SLD used by SAP NetWeaver development infrastructure by following these steps:

1. From the **System Landscape Directory** landing page, click on **Name Reservation**.
2. Choose **Enable This SLD as Name Server for NWDI**.

After the name server has been activated for the SAP NetWeaver development infrastructure development, you can create prefixes in advance. To add a new prefix, follow these steps:

1. From the **System Landscape Directory** landing page, click on **Name Reservation**.
2. Make sure you're in the **Name Prefixes** tab.
3. Click the **New Name Prefix** button. You'll see the screen shown in Figure 3.9.
4. Choose the desired **Name Category**. Because this is for an SAP PO development, choose either **Development Component Name** or **Software Component Name**.

What you choose depends on whether you want to reserve names for a SC or a DC. The concept of DCs is relevant for Java development and will be explored in Chapter 16.

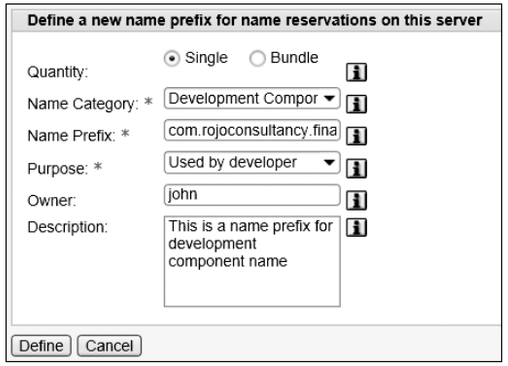


Figure 3.9 Adding a New Name Prefix in the SLD Name Reservation

5. Specify the name prefix. You can specify a name up to 256 characters long. In our example, we used the name **com.rojoconsultancy.finance**. As you can see, this name contains an Internet domain name and department. In this way, you can categorize your namespaces.
6. To complete the task, click **Define**.

Now that you've reserved a name prefix, you can predefine some DCs in the name server and make use of the previously reserved name prefix. To define a name (for a DC or SC), follow these steps:

1. From the **System Landscape Directory** landing page, click on **Name Reservation**.
2. Make sure to be in the **Names** tab.
3. Click the **New Name** button. You'll see the screen shown in Figure 3.10.
4. Select the desired **Name Prefix**. In this case, the previously created **com.rojoconsultancy.finance** is selected.
5. Specify the name of the DC by adding a name after the "/" character.
6. Specify the **Caption** (a short description) for the DC name.
7. Click **Reserve**.

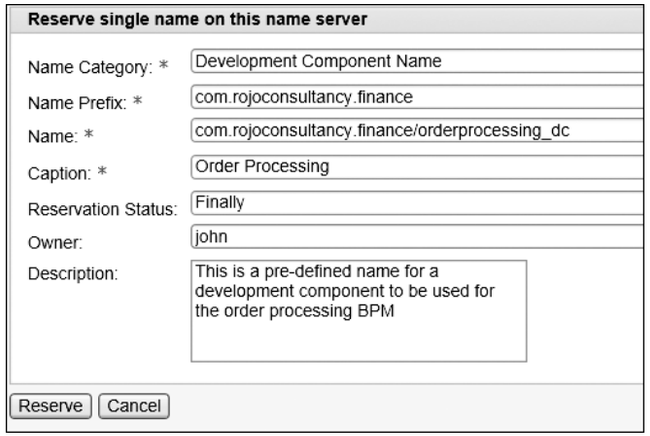


Figure 3.10 Adding a Name in the SLD Name Server

Now, we're ready to use this DC name during development in the SAP NetWeaver development infrastructure.

Common Information Model Instances and Classes

The Common Information Model (CIM) is a standard based on an object-oriented modeling approach. The standard is championed by the Distributed Management Task Force (DMTF). CIM is used to model hardware and software objects and elements. SAP uses the CIM to capture key properties to identify and to specify a product. It contains metadata information such as the name, vendor, description, and caption of a product. The CIM class and CIM instance are two key concepts of the CIM.

A CIM class represents a group of artifacts and objects with similar properties. A CIM class can be associated with one or more other CIM classes. For more information on the CIM standard, visit *www.dmtf.org*.

For the purpose of SAP PO, the CIM is used to hold the description of components, products, and the system landscape. To view and maintain the CIM Instance and classes in your SLD, select the **CIM Instance and Classes** link from the **System Landscape Directory** landing page.

You can update existing descriptions or CIM content in your SLD by using the import functionality. See Section 3.3 to learn more about how to update your existing CIM.

3.2 Registering Systems to the System Landscape Directory

As the previous section explained, it's important to make all participating systems in your SAP landscape register themselves and report to the SLD. To register your system, you need to create a connection between these systems and the SLD. These connections can be achieved using Remote Function Calls (RFCs) or HTTP and are commonly referred to as *data suppliers*. Data suppliers collect a range of data from SAP systems, including the following:

- Application servers and instances
- RFC gateways
- SAP products
- Installed software components
- Installed support packages
- Compound systems
- HTTP ports
- Network services
- liveCaches
- SAP message servers
- RFC destinations
- Hosts
- Clients
- Databases
- CIM associations between generated objects

The collected data is periodically sent to the SLD using CIM over HTTP.

3.2.1 Connecting to ABAP-Based Systems

For ABAP-based systems (from SAP Basis release 6.4.0 and up), it's possible to use data suppliers to send the data to the SLD bridge via an RFC connection and SAP Gateway (as depicted in Figure 3.11). In an ABAP-based system, the data supplier uses data collection programs to collect the needed data. For SAP NetWeaver AS ABAP systems, you can modify data collection programs to change their behavior.

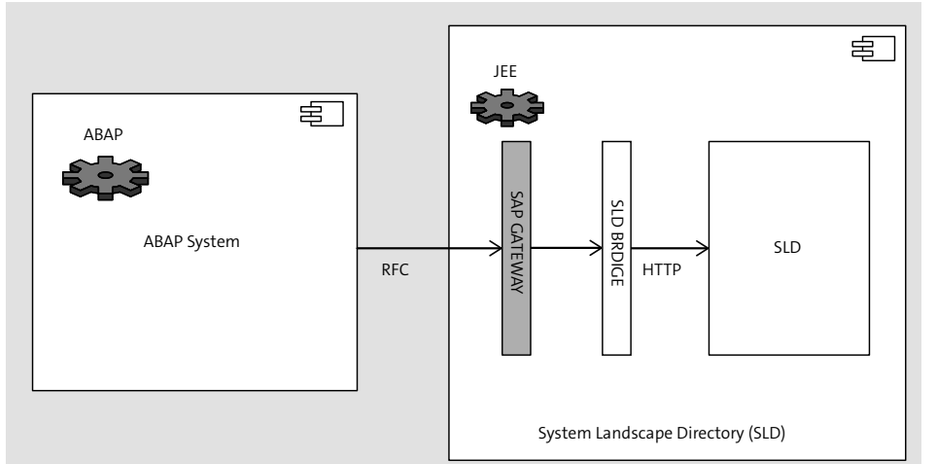


Figure 3.11 Connecting an ABAP System to the SLD

Tip

For an ABAP-based system on SAP Basis release 4.0B and up, use Transaction RZ70 to set up SLD data suppliers.

For SAP systems running on an SAP NetWeaver AS ABAP older than 6.40, you can't configure data suppliers using Transaction RZ70. This configuration will need to be manually performed for such systems.

3.2.2 Connecting to Java-Based Systems

For Java-based systems, configuration of data suppliers is automatically performed during the installation of the system (see Figure 3.12). You can manually change it afterward using SAP NetWeaver Administrator.

To change the data supplier in a Java-based system, follow these steps:

1. Go to the SAP NetWeaver Administrator via *http://<hostname>:<port>/nwa*.
2. Navigate to **Configuration • Security • Destinations**.
3. Change or create a destination with the name "SLD_DataSupplier" and of type **HTTP**.

4. You can now enter or change the details in the **URL** field to point to the SLD system. Don't forget to enter the login details under the **Logon Data** tab.
5. Test your destination by using the **Ping Destination** button.

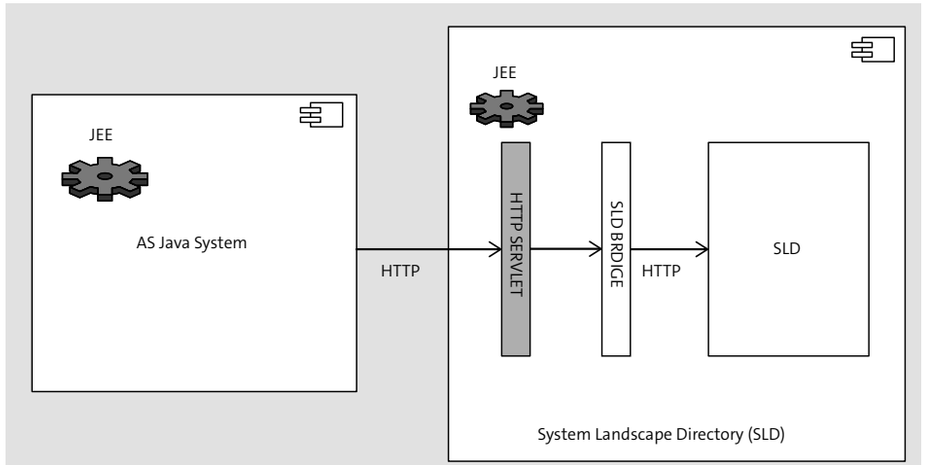


Figure 3.12 Connecting an SAP NetWeaver AS Java System to the SLD

You also have the option to force an SAP NetWeaver AS Java system to collect and send the data to the SLD server at any given time by performing the following steps:

1. Go to the SAP NetWeaver Administrator via *http://<hostname>:<port>/nwa*.
2. Navigate to **Configuration • Infrastructure • SLD Data Supplier Configuration**.
3. Click on the **Collect and Send Data** button (see Figure 3.13).

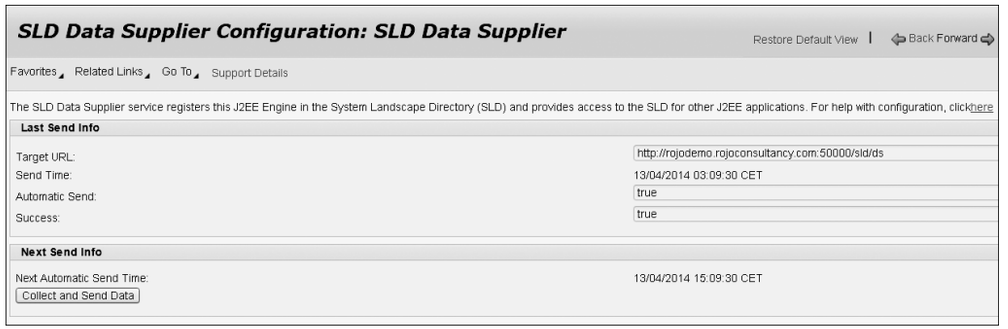


Figure 3.13 Forcing an SAP NetWeaver AS Java System to Push Data to the SLD

3.2.3 Connecting to Other Systems

Systems other than ABAP or those based on SAP NetWeaver AS Java can also be connected to the SLD and can send their data using an executable program called `sldreg`. See Table 3.2 for a summary of the approaches to follow to connect a given application system to the SLD, depending on its type and version.

System type	Approach
SAP system with SAP NetWeaver AS ABAP older than 6.40	Manually update SLD data.
SAP System with SAP NetWeaver AS ABAP equal to or greater than 6.40	Configure data suppliers to automatically update SLD using Transaction RZ70.
SAP system with SAP NetWeaver AS Java older than 7.1	Configure via the Visual Administrator tool.
SAP system with SAP NetWeaver AS Java equal to or greater than 7.1	Configure via the SAP NetWeaver Administrator tool.
Standalone and third party	Manually update SLD data.

Table 3.2 Overview of Approaches to Connect a System to the SLD

The next section discusses the components and available features of the SLD.

3.3 Administration of the System Landscape Directory

In this section, we'll explore some administrative functionalities of the SLD, focusing on those functionalities that are commonly used and relevant to an SAP PO installation. The main page used for the administration of your SLD can be reached by clicking on the **Administration** menu link from the **System Landscape Directory** landing page. From this page, you can distinguish the server and content categories, which will be explored in this section.

3.3.1 Server

This category contains applications that expose different aspects of the SLD server. We'll explore each application.

Status of the System Landscape Directory

To view the current status of the SLD, whether it's currently running or not, check the traffic light on the top left of the SLD **Administration** screen (see Figure 3.14).

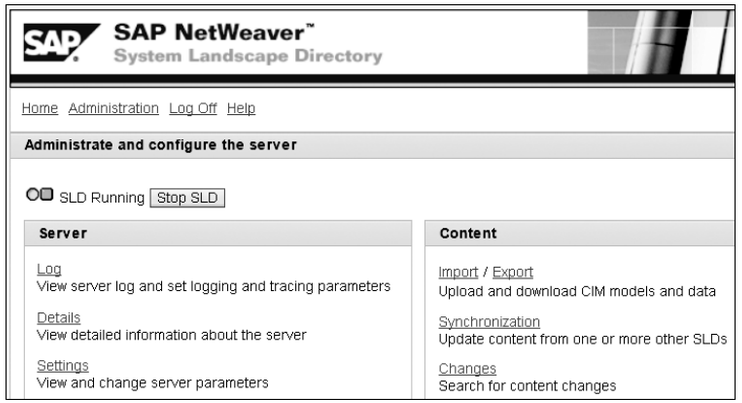


Figure 3.14 SLD Administration Screen

Like any conventional traffic light, a green light will mean that your SLD is up and kicking, whereas a red light will indicate that it's down and might require manual action.

Stop or Start the System Landscape Directory

You can temporarily stop or start the SLD using either the **Stop SLD** or **Start SLD** button, located at the right side of the traffic light (refer to Figure 3.14).

Log

With this logging functionality, you can view the SLD server logs. From this page, for example, you could get more details about failing data imports or exports. It's also possible to set logging and tracing parameters here.

Details

From the **Details** page, it's possible to see all detailed information about your SLD server. As shown in Figure 3.15, the following details are displayed:

- **General**
  This tab contains details of the SLD such as running statuses, versions, SAP Java Virtual Machine (SAP JVM) memory, JVM version, Java system properties, and so on.

- **Data**
  This tab contains CIM-related totals and information such as counts of CIM classes, instances, and associations.
- **Data Suppliers**
  This tab has details about the data supplier and which other SLDs are being updated by the local SLD. Figure 3.15 shows that the local SLD server is forwarding data to another (central) SLD.
- **Server**
  This tab has information such as CIM service version, utilization capacity, hit rate, and so on.

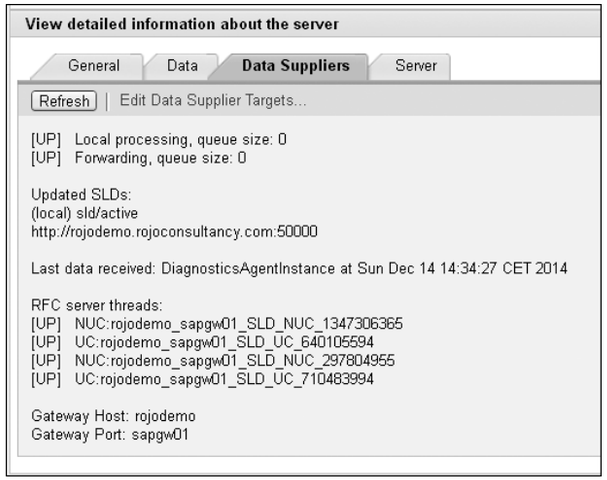


Figure 3.15 SLD Administration Details Page and Available Information

3.3.2 Data and Content

Data and content can be added to the SLD via three different options: data import, system self-registration, and manual creation of data.

Import and Export

This functionality enables the import of the CIM and data from a local file. You can also export the CIM and data of the current SLD via the export functionality.

You can use the export function to perform a full backup of the current SLD data. It's also possible to perform an incremental export. Incremental exports only export the

delta objects that were added to the SLD since the last time an export happened. As such, the incremental method of performing an export requires good administration.

Synchronization

If you have multiple SLDs, and you need to keep their content in sync, then you might opt to synchronize them or to transport the content between the different SLDs. It's then possible to choose between a fully automated synchronization and a manual synchronization. The synchronization of SLD content will be broadly explored in Section 3.4, when we discuss the different SLD strategies. During a typical synchronization, the data and content from one source SLD are propagated to another target SLD.

3.4 System Landscape Directory Strategies

Given the role played by the SLD, it's important to come up with a proper SLD strategy to be used in your organization. There are many approaches to run an SLD in your landscape. To start with, ask yourself if you need one or multiple SLDs in the landscape. To properly answer this question, you need to look at the organization's landscape and requirements. Consider the following questions:

- How big is the landscape, and how many systems will be using the SLD?
- How spread out are these systems? Are all the systems part of the same network, or are they spread across the globe?
- Do you want to keep the data of your different environments or stages separated?

Depending on your answers to these questions, you may want to consider multiple SLDs. Having more than one SLD can bring some advantages:

- Data views are separated (the administrator of each group can only see a limited view or subset of data).
- Your SLD data and content are highly available.
- Any changes to a local SLD's data will only impact a limited part of the organization. Because of the separation of SLDs, it becomes possible to choose the right time to reflect the changes in the rest of the organization (e.g., after a successful test of the performed changes and after you feel comfortable reflecting the changes in the rest of the organization).

With more than one SLD instance in your landscape, the need arises to distribute the data between the different SLDs and to set up a synchronization mechanism between them. There are different approaches to synchronize data between the SLDs, including the following:

- Manual export and import of data (full or delta) using the enhanced Change and Transport System (CTS+)
- Automatic forwarding of data suppliers
- Full automatic synchronization

In the next sections, we'll explore each of these approaches.

3.4.1 Manual Export and Import of Data

If you facilitate the manual export of any selected data from your SLD, then the exported data can be manually imported into the next environment or local SLD. This approach has the major downside of requiring human manual action. It's better to couple this approach with a CTS+. To manually export an SLD object (using a business system here as an example), follow these steps:

1. From the **System Landscape Directory** landing page, select **Business Systems**.
2. Filter and select the business system that you would like to export.
3. With the business system selected, click on the **Export** button (see Figure 3.16 ❶).

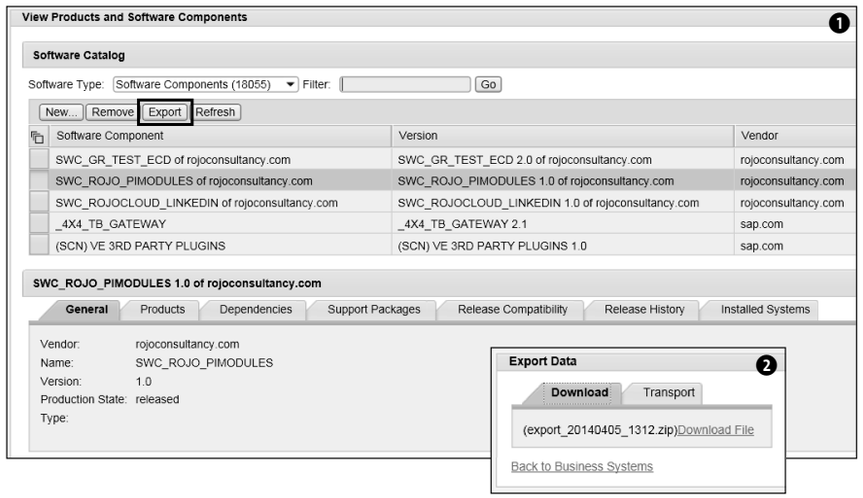


Figure 3.16 Steps to Manually Export an SLD to a Business System

4. Click on **Download File** from the **Download** tab ❷, and save the resulting TPZ file on your local file system.
5. The saved TPZ file can then be imported into the target SLD system.

3.4.2 Automatic Bridge Forwarding

The automatic bridge forwarding approach copies all technical systems data in the source SLD and distributes them to the next SLD system. Most of the time, this approach is used to synchronize the central or master SLD with local SLDs. It's also referred to as one-way distribution or SLD bridge forwarding. This approach only forwards data that is received in the SLD via a data supplier. The manually entered data isn't propagated. To set up automatic bridge forwarding, follow these steps:

1. From the **System Landscape Directory** landing page, select **Administration** and **Data Suppliers**.
2. Under **Update Other SLDs**, click on the **Add** button (Figure 3.17 ❶).
3. Fill in the details of the target SLD system ❷.

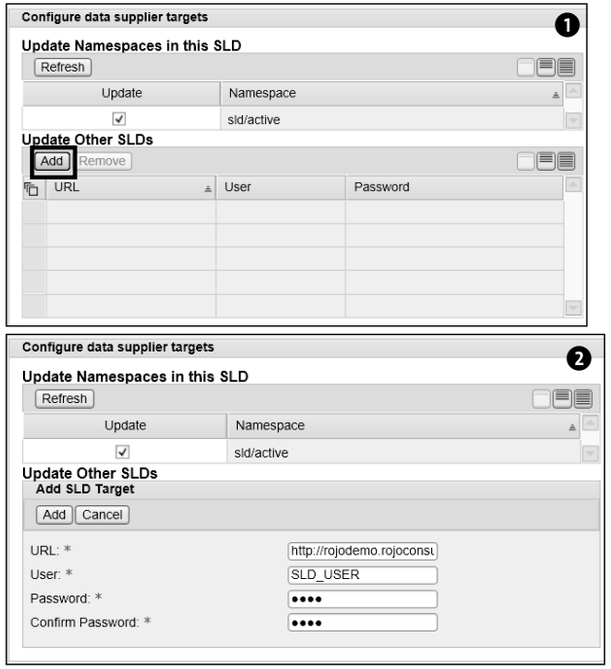


Figure 3.17 Steps to Set Up an SLD Automatic Bridge Forwarding

3.4.3 Full Automatic Synchronization

The automatic synchronization mechanism ensures that all data entered manually in the local SLD and updated by the data supplier is automatically replicated and pushed to the target SLDs. This is a favorable mechanism because it reduces the need to manually keep the SLDs in sync and therefore reduces the risk of human error. It's important to note that the automatic synchronization is supported as of SAP NetWeaver AS Java 7.1 SPS 1 and higher. A synchronization can be unidirectional or bidirectional:

- **Unidirectional**
  Data from the source SLD is synchronized with the target system. This implies that all the data present in the source SLD is also present in the target SLD, but the target SLD might have more data.
- **Bidirectional**
  In this situation, both SLDs are identical. The SLDs can be seen as each other's clones, in terms of their data. Bidirectional SLD synchronization can be used as a way to create an SLD backup or redundancy (see Figure 3.18).

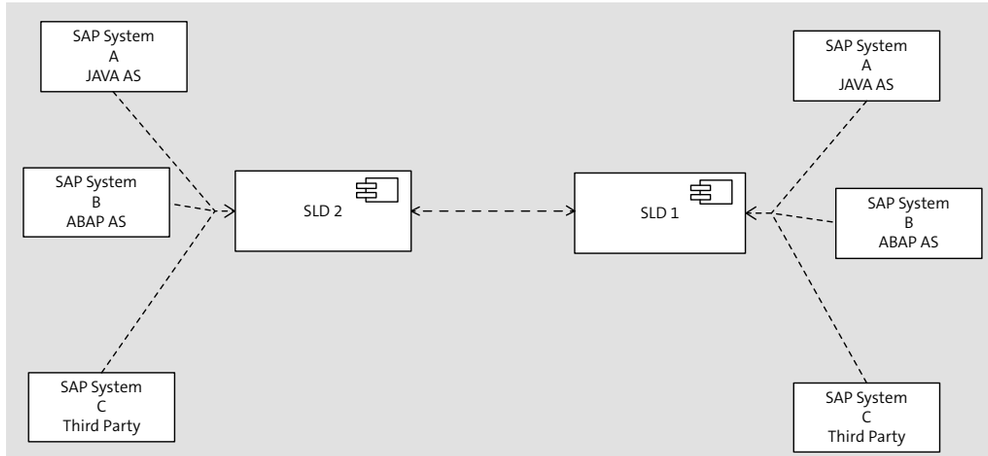


Figure 3.18 Bidirectional Full Automatic Synchronization Example

The automatic synchronization uses HTTP and is asynchronous. To configure the full automatic synchronization of your SLD, follow these steps:

1. From the **System Landscape Directory** landing page, choose the **Administration** link on the top menu and then the **Synchronization** link in the resulting page.

2. You're then forwarded to a **Maintain SLD Content Synchronization** screen. Click the **Add** button (see Figure 3.19 ❶).
3. In the next screen, select the following values:
   - For the **Source**, select the **Remote (Sync with another SLD)** radio button ❷.
   - Make sure the **Bidirectional** checkbox is selected.
4. In the subsequent screen, maintain the hostname and logon details to be used to connect to the target SLD.

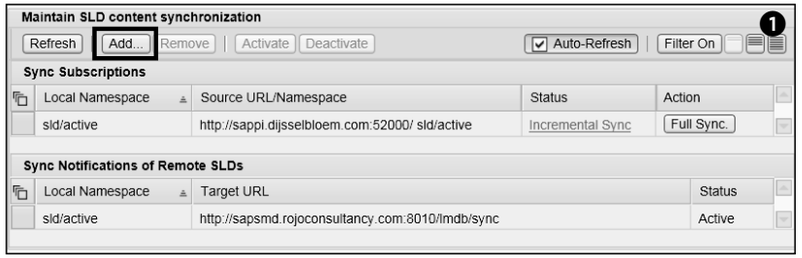


Figure 3.19 Steps to Configure the SLD Full Automatic Synchronization

After you've finished providing details in the second step of the wizard, follow these steps:

1. Define the source namespace and rank. The source namespace represents the namespace from the source SLD to be synchronized. By default, the **sld/active** namespace is selected. However, you can choose another namespace from the dropdown values. Follow the wizard's steps.

   The rank helps resolve conflicts. The namespace rank number specifies the priorities of SLD data in the synchronization process. A higher rank number means that the data of the source SLD namespace will "win" in conflicts with data of the target SLD's namespace (see Figure 3.20 ❶).

2. Maintain the URL and login details of the local SLD. In addition, you need to choose the target namespace and its rank.
3. You'll be presented with a summary of all the values you entered. Click **Finish** after checking that the summary is correct.
4. You can now check the status of the full sync by selecting the **Full Sync (Active)** link. You can also force a full sync by clicking on the **Full Sync** button ❷.

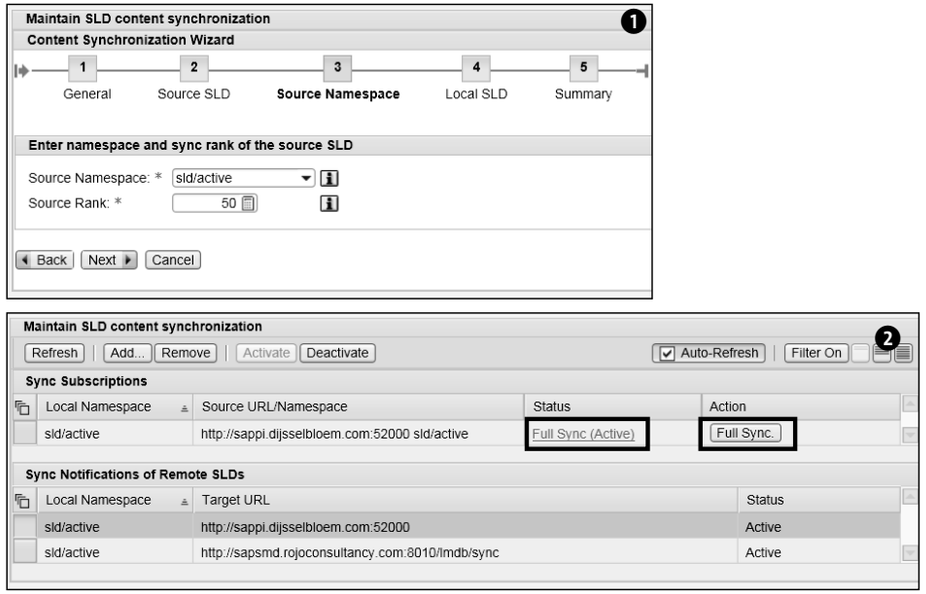


Figure 3.20 Configuring the SLD Full Automatic Synchronization: Defining Ranks

3.5 Tips and Tricks

In this section, we'll share some tips and step-by-step guides to perform some administrative tasks in the SLD.

3.5.1 Naming Convention

Consider using a convenient naming convention when naming your SLD objects. A good naming convention has numerous advantages, including the following:

- **Creates order**
  When the objects are named in a logical manner, their purposes will be obvious to every developer. When looking at the name of an object, it should be obvious which type of SLD object you're looking at, and you should have an idea of what role it plays in the landscape.
- **Creates a recognizable structure**
  A good naming convention makes life easier for the developers and people taking care of the operational side of the SLD, especially when you're dealing with a complex landscape and a growing number of systems and interfaces.
- **Increases productivity**
  When the naming convention is simple and logical, it becomes easier to group, sort, find, and search objects, which will help speed up and simplify the development process. This in turn will result in an easy adoption of the naming convention and increased productivity.

Having a naming convention alone won't cut it. It's important to encourage everyone to make a good habit of using the naming convention and to facilitate its adoption. Furthermore, someone needs to be responsible for enforcing the naming convention.

3.5.2 Keeping Your System Landscape Directory Catalogs Up to Date

It's strongly advised to always keep the SAP-provided catalogs of content up to date. You can update the SLD catalog content by following these steps:

1. Download SAP content from the SAP Service Marketplace via *http://service.sap.com/swdc*.
2. Navigate to **SAP Software Distribution Center** • **Download** • **Support Packages and Patches** • **Entry by Application Group** • **SAP Technology Components** • **SAP CR Content**. You can then download the relevant content (provided as ZIP files) and place it into a local file system. Be aware that you don't need to download the entire software catalog. Each package is a delta of the previous version, so you can download all the packages subsequent to your current version and import them in order.
3. Import the files into the SLD. From the **System Landscape Directory** landing page, go to **Administration** • **Import** (under **Content**).
4. Select the ZIP file via the **Browse** button.

5. Click the **Import** button. Depending on the size of the file to be imported, you can expect this update process to run for a while (see Figure 3.21).

> **Note**
>
> If you need to import multiple CIM files, make sure that you import them in the right order and sequence because each package is a delta of the previous version.

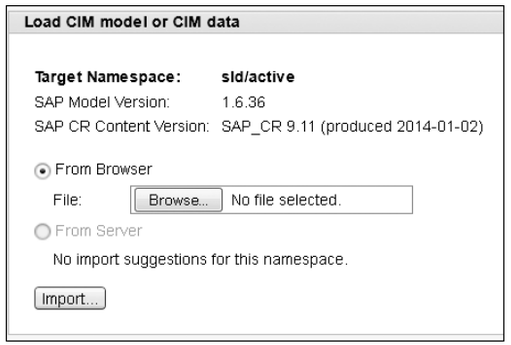


Figure 3.21 Import a New SLD CIM or Data

Figure 3.21 shows the CIM and data import functionality. On this screen, you can also see the version of the content currently loaded in the SLD. For obvious reasons, it's important to verify the current CIM version to determine the version to be imported.

3.5.3 Self-Registration of a Java-Based System with the SLD

To let an SAP NetWeaver AS Java-based system register its technical system with the SLD, you need to configure its data supplier. To configure data suppliers of the SAP NetWeaver AS Java-based system, perform the following tasks:

1. Log in to the SAP NetWeaver Administrator of the concerned system, via *https://<hostame>:<port>/nwa*.
2. Go to the **Destinations** section by following the path **Configuration • Connectivity • Destinations**.
3. Filter for and select the **SLD_DataSupplier** destination, and edit it (see Figure 3.22).
4. Change the URL field (in the **Connection and Transport** tab) to show the URL of the SLD that the system needs to register itself with. Note that the format of the URL should be *http://<hostname>:<port>*.

5. Add the required login details in the **Logon Data** tab.
6. Test your destination setup by clicking on **Ping Destination** (see Figure 3.22). After a successful ping, you're ready to register the system with the SLD and send some data.

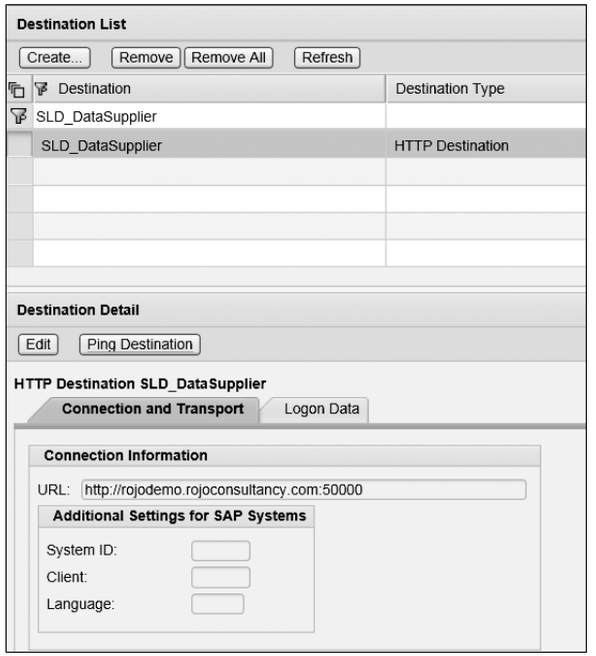


Figure 3.22 Steps to Self-Register a Java-Based System with the SLD

To manually push the data to the SLD or to force the self-registration of an SAP NetWeaver AS Java system, follow these steps:

1. From SAP NetWeaver Administrator, navigate to **Configuration • Infrastructure • SLD Data Supplier Configuration**.
2. You should see information about the next scheduled or automatic send time. Select the **Collect and Send Data** option. If everything was well configured, then you'll get a successful send data message.
3. You should now be able to log in to the target SLD and verify that the new system registered itself. You can find this out by looking up the list of technical systems in the target SLD.

3.5.4 Configuring Data Suppliers from One SLD to Another

All data about technical systems that an SLD receives can be forwarded to another SLD using the data supplier. It's important to stress that only data which arrived via data suppliers (such as technical systems) are forwarded. All manually created data (such as business systems) aren't included in the forwarded data. Use the manual export and import options described in the next section for manually created SLD data.

To set up data forwarding between a source and target SLD (or data supplier target), perform the same steps as described previously for automatic bridge forwarding (Section 3.4.2; also refer to Figure 3.17).

3.5.5 Manual Export and Import of Data

It's possible to manually export any data from a source SLD to a target SLD by using the import and export features. In this way, you could export and import manually created business systems, for instance. The steps to export a business system were described previously in Section 3.4.1. We'll extend those steps by mentioning the required steps to import an exported business system into a target SLD.

The required steps are as follows:

1. From the **System Landscape Directory** landing page of your target SLD, click on the **Administration** link.
2. Then click on **Import.**
3. You're offered the possibility to import an archive file. Choose the file that you exported from the source SLD.
4. Start the import procedure by clicking on the **Import** button.
5. You'll see another screen that requires you to confirm or cancel your import. Click **Continue Import.**
6. After this step, a message confirming the successful import of the new business system will be displayed.

3.5.6 Connecting the SLD to CTS+ to Facilitate the Export and Import of SLD Data

To facilitate the propagation of SLD data from one SLD to another, we recommend that you use a transport mechanism rather than the manual export and import feature.

CTS+ is the recommended tool to manage your SAP PO transports. To connect your development SLD to CTS+, follow these steps:

1. Make sure that you have a communication user in the SAP NetWeaver AS Java that belongs to the `SAP_SLD_Administrator` group. If such a user doesn't exist, then create it before proceeding to step 2.
2. Go to the SAP NetWeaver Administrator via *http://<hostname>:<port>/nwa*.
3. Navigate to **Configuration • Connectivity • Destinations**.
4. Create a destination called "sap.com/com.sap.tc.di.CTSserver" if it doesn't already exist. Give it the type **RFC,** and click **Next**. Note that the destination name is predefined by SAP and therefore must match exactly. It's also case sensitive (see Figure 3.23 ❶).
5. In the next wizard step, add details of the transport system, including logon details to be used to connect to it. Make sure to use a technical user for the logon data ❷.
6. Click **Finish**.

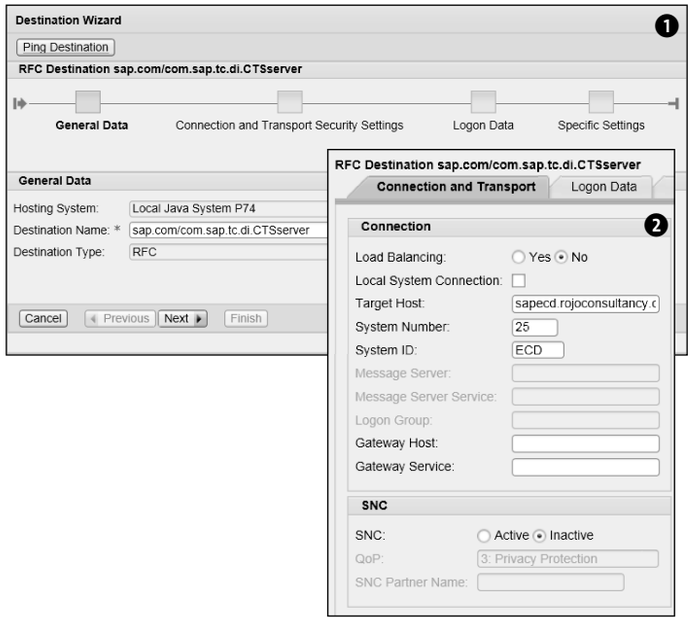


Figure 3.23 Steps to Set Up a Destination from SAP PO to CTS+

Now that you've connected your system to CTS+, you can test the export functionality via the following steps:

1. Log in to your SLD, and navigate to the object you want to export via CTS+. As an example, export a SC.
2. From the **System Landscape Directory** home or landing page, select **Software Components**.
3. Select the SCs in question, and click on **Export**.
4. You'll see an **Export** screen with the **Transport** tab enabled. If the steps for configuring the destination to CTS+ aren't performed properly, you'll see an error here, and the **Transport** tab won't be enabled (see Figure 3.24).

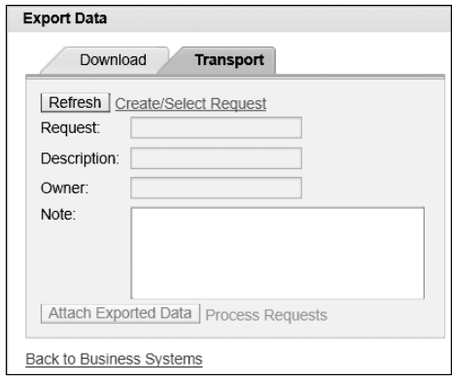


Figure 3.24 Exporting SLD Objects with CTS+ Enabled

3.6 Exercise: Configuring the System Landscape Directory

This exercise is meant to serve as an implementation example and provides a practical insight into the SLD and its objects. We begin with a description of the exercise, briefly touch on what the solution would look like, and then provide step-by-step instructions for the solution.

3.6.1 Exercise Description

Imagine that you're employed by a fictitious company called Reader's Paradise as an SAP integration expert. Reader's Paradise is a bookstore that specializes in selling exclusive books. The company uses SAP PO, and you've been asked to build an interface enabling Reader's Paradise to order books from its main supplier, Book Fountain. This means that you need to build an interface that will fetch a purchase order from Reader's Paradise's system and send the order to Book Fountain.

This first exercise focuses on creating all necessary SLD objects required to build the interface, as illustrated in Figure 3.25.

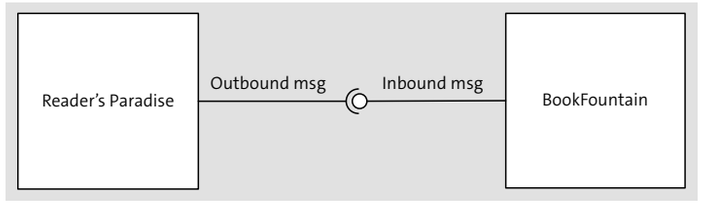


Figure 3.25 Scenario to be Built for Purchase Order Transfer

3.6.2 Exercise Solution Approach

In this first exercise, you're only required to create the relevant SLD objects. The SLD objects to be created include the following artifacts:

- A product
- Three SCs (given that we want to use the three components model, the SC containing the shared object needs to have a dependency to the other two SCs)
- Two technical systems (for sender and receiver of the third-party type)
- Two business systems (for sender and receiver of the third-party type)

A graphical representation of these artifacts and the relationships among them is depicted in Figure 3.26.

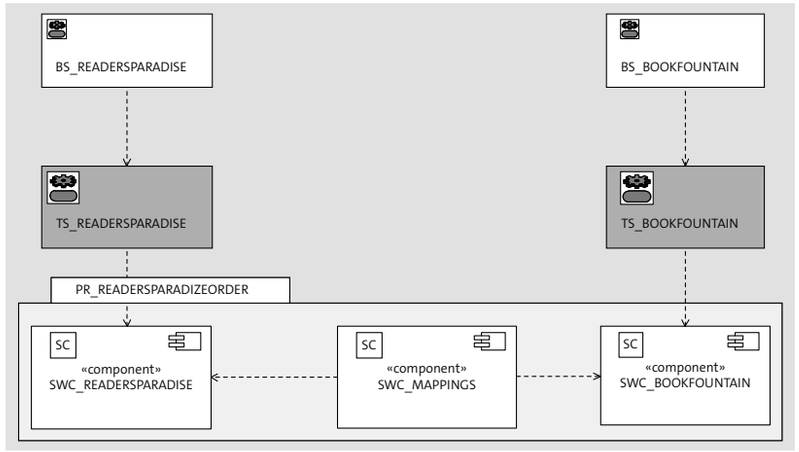


Figure 3.26 The SLD Artifacts and Their Relationships

We provide the step-by-step solution of the exercise in the next section.

3.6.3 Exercise Step-by-Step Solution

The activities and steps to create the objects (as listed in the exercise solution approach) are described in the following subsections.

Log On to the System Landscape Directory

To log on to the SLD, open a browser, go to *http://<hostname>:<port>/sld*, and log in with your credentials when prompted to do so. In the preceding URL, *<hostname>* is the hostname of the SAP NetWeaver system, and *<port>* is the port number of the SAP NetWeaver system.

Create a Product

To create a product that will hold the SCs, follow these steps:

1. Click on the **Products** tab of the **System Landscape Directory** landing page.
2. Click on the **New** button to follow the wizard for creating a product (see Figure 3.27).

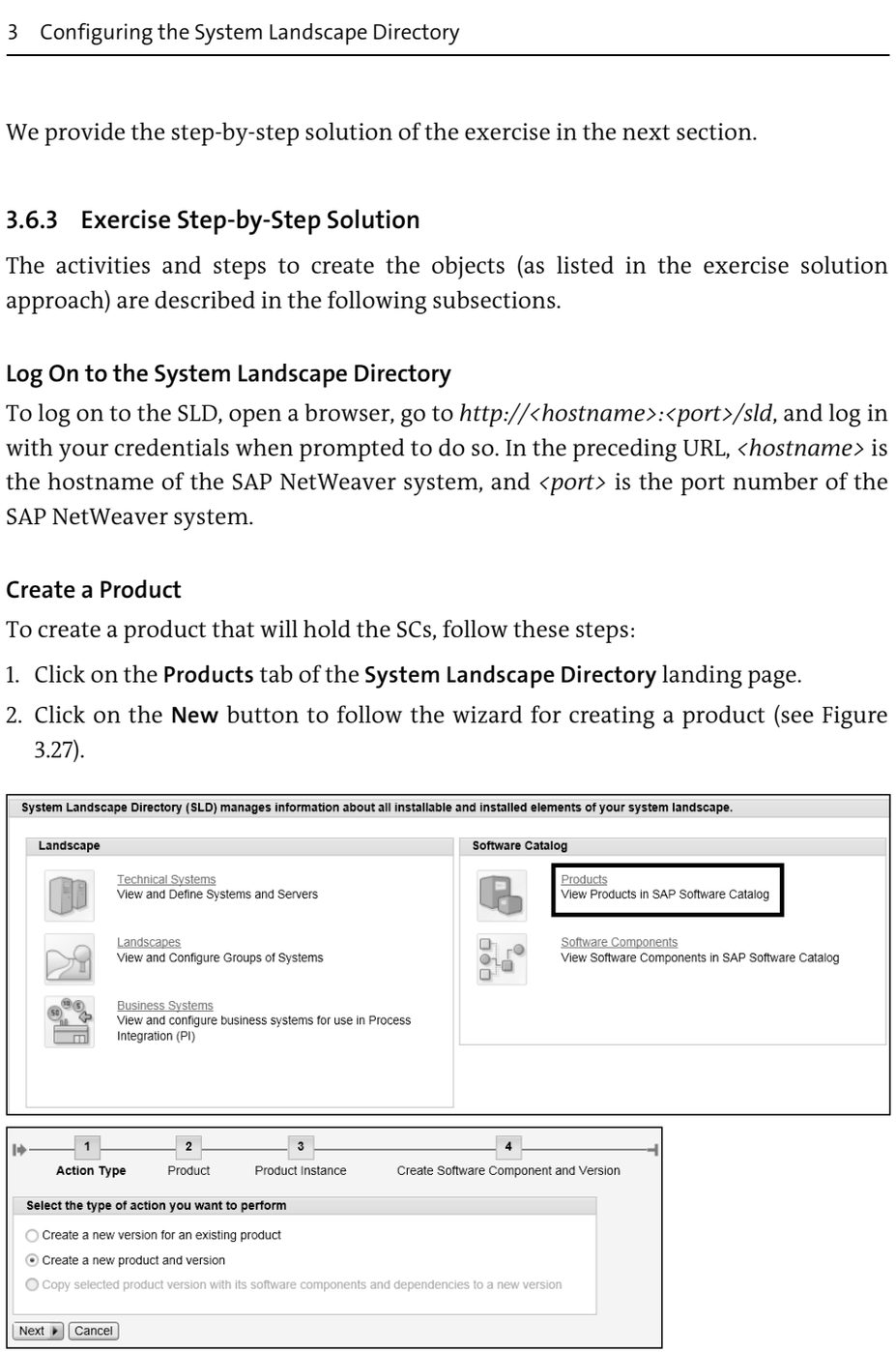


Figure 3.27 First Steps for Creating a Product in the SLD

3. Follow the wizard, and enter the name "PR_READERSPARADISE" for your product. Enter "ReadersParadise.com" as the product vendor (see Figure 3.28).
4. Click on **Next**, and you'll be asked to fill in data to create the SC.

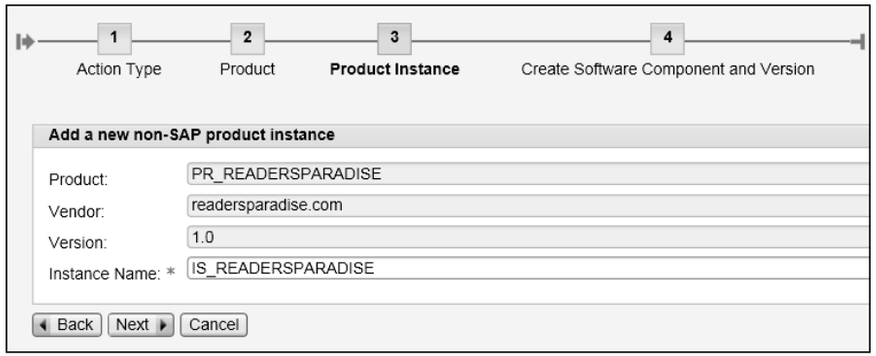


Figure 3.28 Creating a Product in the SLD

Create Three Software Components

Because we're following the three SC model, we'll need three separate SCs to hold the sender, receiver, and canonical objects.

To create the SCs that will hold the objects and artifacts during design time and with which the technical and business systems will be associated, follow these steps:

1. Click on the **Software components** tab of the **System Landscape Directory** landing page.
2. Click on the **New** button to follow the wizard for creating a new SC. Follow the wizard, and name the SC "SWC_READERSPARADISE". The final results are shown in Figure 3.29.
3. Click on **Finish**.
4. Repeat steps 2 and 3 to create two more SCs, with the names "SWC_BOOKFOUNTAIN" and "SWC_MAPPINGS".

> **Note**
>
> You need to select the product (**PR_READERSPARADISE**) that was previously created for your SCs.

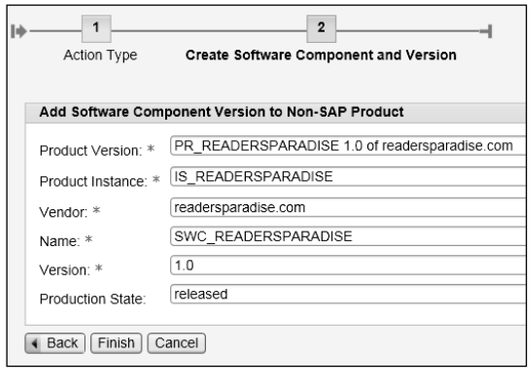


Figure 3.29 Creating a Software Component in the SLD

Given that the SC named SWC_MAPPINGS will contain the mapping and transformation object, it needs to have dependencies to both SWC_READERSPARADISE and SWC_BOOKFOUNTAIN SCs. To create dependencies for the SWC_MAPPINGS, follow these steps:

1. From the **System Landscape Directory** landing page, select **Software Components**.
2. Search or filter for the SC to which the dependencies need to added—in this case, the SC called SWC_MAPPINGS.
3. Click on the **Dependency** tab, and filter for or select the SC SWC_READERSPARADISE.
4. Click on **Define as Prerequisite Software Components**. The final result should look like Figure 3.30.

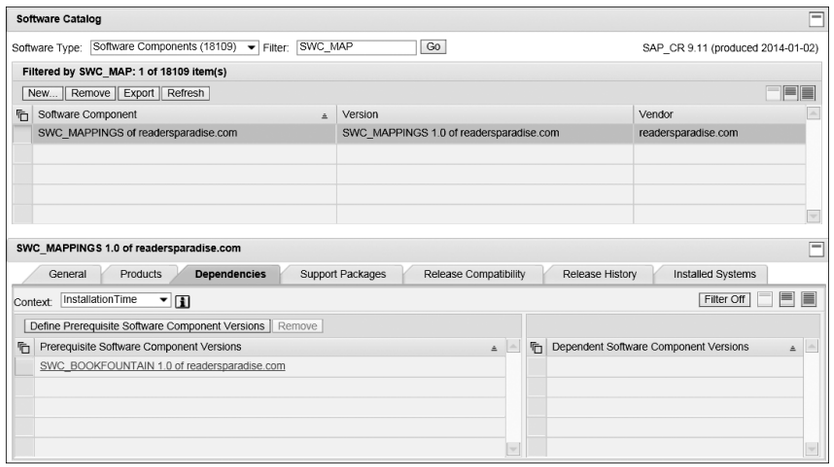


Figure 3.30 Creating Software Component Dependency to Other Software Components

5. Repeat steps 2, 3, and 4 to define another dependency from SWC_MAPPINGS to SWC_BOOKFOUNTAIN.

The dependencies are defined to ensure that the Enterprise Services Repository (ES Repository) objects (data, messages types, and service interfaces) of SWC_BOOKFOUNTAIN and SWC_READERSPARADISE can be accessed from the SC SWC_MAPPINGS.

Create Technical Systems (Sender and Receiver)

To create a technical system with which the business system will be associated, follow these steps:

1. Click on the **Technical Systems** tab of the **System Landscape Directory** landing page.
2. Click on the **New Technical System** button to follow the wizard for creating a new technical system.
3. Follow the wizard, and specify the type of the technical system as **Third Party**.
4. Click on the **Next** button to proceed to the next screen.
5. Name the technical system "TS_READERSPARADISE" to represent the sending system (see Figure 3.31).

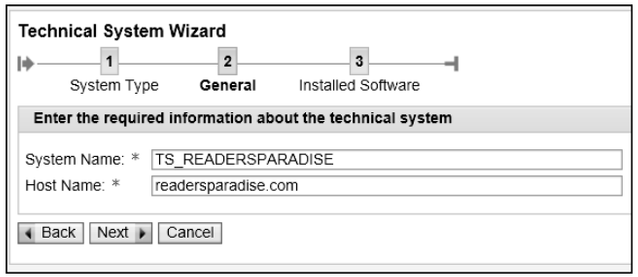


Figure 3.31 Creation of a Technical System

6. Select the products and SCs that are installed in this technical system. If you don't add a SC to a technical system, then it won't be possible to use its service interfaces in your scenario (see Figure 3.32).
7. Repeat steps 2, 3, and 4 to create a new technical system for the receiver system called TS_BOOKFOUNTAIN.

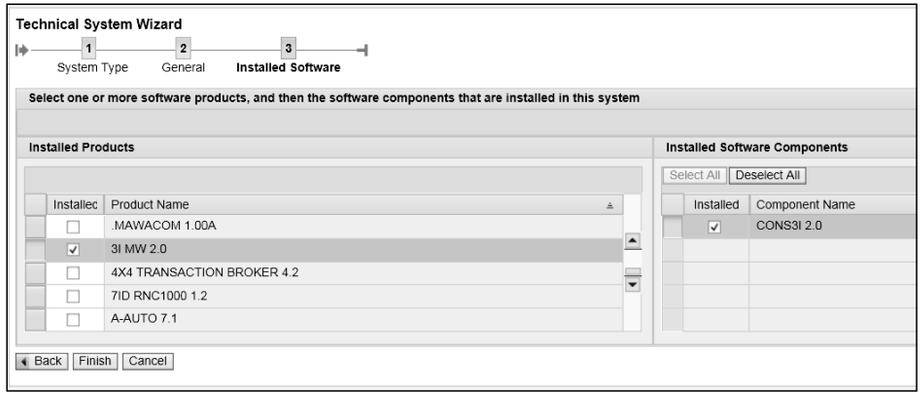


Figure 3.32 Adding Software Components to a Technical System

Create Two Business Systems (Sender and Receiver)

To create a business system that will represent the sender and receiver applications during configuration in the Integration Directory (see Chapter 5), follow these steps:

1. Click on the **Business Systems** tab of the **System Landscape Directory** landing page.
2. Click on the **New Business System** button to follow the wizard for creating a new business system. Click on **Next**.
3. Follow the wizard, and specify the **Type** of business system to be **Third Party/Other**. Click on **Next**.
4. On the next screen, choose the technical system for this business system. Choose **TS_READERSPARADISE** because you're required to create a relationship between a business system and its corresponding technical system. Click on **Next**.
5. On the next screen, name the business system "BS_READERSPARADISE" to represent the sending system. Click on **Next**.
6. Because you linked the new business system to an existing technical system, all products and SCs used in that technical system will be automatically loaded. You still have the option to select or deselect products and SCs on this screen.
7. In the final step, select the Integration Server (SAP PO system) to be used by this application system. You can select from a dropdown list of all SAP PO/SAP PI systems in your landscape.
8. Repeat steps 1 to 7 for a new business system for the receiving system. This last business system needs to be named `BS_BOOKFOUNTAIN` and will need to be linked to the technical system previously created and named `TS_BOOKFOUNTAIN`.

3.7 Summary

In this chapter, we demonstrated that the SLD is a critical component of your landscape, and we explored how it supports the SAP PO middleware by providing it with all necessary foundation objects. This chapter also explored all of the administrative tasks provided by the SLD. These administrative tasks include stopping and starting the SLD, exporting and importing SLD objects, configuring data suppliers, and viewing logs.

Given the critical role that the SLD plays in your organization, the chapter also explored different SLD strategies and approaches to take into consideration during your implementation.

After reading this chapter, you've obtained the foundation required to start building ES Repository objects, a topic that we'll cover in the next chapter.

Contents

Foreword 23
Acknowledgments 25
Introduction 27

PART I Getting Started

1 Introduction to SAP Process Orchestration 33

1.1 Historical Overview and Evolution 33
1.1.1 SAP Process Integration 34
1.1.2 SAP Composition Environment 35
1.1.3 SAP Process Orchestration 36
1.1.4 SAP Process Orchestration 7.5 Highlights 37

1.2 SAP Process Orchestration Components 38
1.2.1 SAP Process Integration 38
1.2.2 Business Process Management 42
1.2.3 Business Rules Management 43

1.3 Architectural Overview and Positioning 45
1.3.1 SAP Process Orchestration Positioning 45
1.3.2 SAP NetWeaver Application Server for Java: System Architecture 47
1.3.3 SAP NetWeaver AS Java System Logical Layers 50

1.4 Installation Options 53
1.4.1 Case 1: SAP Process Integration Dual Stack 53
1.4.2 Case 2: SAP PI Dual Stack and SAP Composition Environment in Separate Installations 54
1.4.3 Case 3: SAP PI Single Stack and SAP Composition Environment in Separate Installations 56
1.4.4 Case 4: SAP PO—SAP PI and SAP Composition Environment in a Single Installation 56

1.5 Summary 57

2 Administration and Development Tools 59

2.1 SAP Process Orchestration Tools 59
2.1.1 Enterprise Services Repository 61
2.1.2 Integration Directory 70
2.1.3 System Landscape Directory 73
2.1.4 Configuration and Monitoring 77

2.2 SAP NetWeaver Administrator 78
2.2.1 Availability and Performance 80
2.2.2 Operations 82
2.2.3 Configuration 82
2.2.4 Troubleshooting 84
2.2.5 SOA 84

2.3 SAP NetWeaver Developer Studio 86
2.3.1 Use and Download 86
2.3.2 SAP Process Orchestration and SAP NetWeaver Developer Studio 87

2.4 Summary 90

PART II Advanced Adapter Engine Extended (AEX)

3 Configuring the System Landscape Directory 93

3.1 System Landscape Directory Components and Features 94
3.1.1 Landscape 94
3.1.2 Software Catalog 100
3.1.3 Development 103

3.2 Registering Systems to the System Landscape Directory 107
3.2.1 Connecting to ABAP-Based Systems 107
3.2.2 Connecting to Java-Based Systems 108
3.2.3 Connecting to Other Systems 110

3.3 Administration of the System Landscape Directory 110
3.3.1 Server 110
3.3.2 Data and Content 112

3.4 System Landscape Directory Strategies 113
3.4.1 Manual Export and Import of Data 114
3.4.2 Automatic Bridge Forwarding 115
3.4.3 Full Automatic Synchronization 116

3.5 Tips and Tricks 118
3.5.1 Naming Convention 118
3.5.2 Keeping Your System Landscape Directory Catalogs Up to Date 119
3.5.3 Self-Registration of a Java-Based System with the SLD 120
3.5.4 Configuring Data Suppliers from One SLD to Another 122
3.5.5 Manual Export and Import of Data 122
3.5.6 Connecting the SLD to CTS+ to Facilitate the Export and Import of SLD Data 122

3.6 Exercise: Configuring the System Landscape Directory 124
3.6.1 Exercise Description 124
3.6.2 Exercise Solution Approach 125
3.6.3 Exercise Step-by-Step Solution 126

3.7 Summary 131

4 Working with the Enterprise Services Repository and Registry 133

4.1 Basic ES Repository Technical Concepts 134
4.1.1 Functional Blocks 134
4.1.2 First Steps in the Enterprise Services Repository 136
4.1.3 Service Interface 137
4.1.4 Integration Patterns: Stateful and Stateless Communication 138
4.1.5 Asynchronous versus Synchronous 139
4.1.6 Quality of Service 141

4.2 Design Objects 141
4.2.1 Software Component Versions 142
4.2.2 Folders 142
4.2.3 Namespaces 142
4.2.4 Mappings 144

4.2.5 Process Integration Scenario 147
4.2.6 Actions 148

4.3 Data Types and Message Types 149
4.3.1 Data Types 149
4.3.2 External Definitions 149
4.3.3 Message Type 150
4.3.4 Additional Design Objects 151

4.4 Exercise: Working with the ES Repository and Registry 153
4.4.1 Exercise Description 153
4.4.2 Exercise Solution Approach 153
4.4.3 Exercise Step-by-Step Solution 154

4.5 Summary 169

5 Working with the Integration Directory 171

5.1 Integration Directory Overview 171

5.2 Collaboration Profiles 173
5.2.1 Party 173
5.2.2 Communication Component 174
5.2.3 Communication Channel 177
5.2.4 Communication Component without a Party 186

5.3 Adapter Types 189
5.3.1 Technical Adapters to Enable Communication with SAP or Third-Party Systems 189
5.3.2 Application Adapters to Enable Communication with an SAP System 190
5.3.3 SAP Industry Adapters 191
5.3.4 Third-Party-Developed Adapters 191

5.4 Integrated Configuration 192
5.4.1 Inbound Processing 193
5.4.2 Receiver 195
5.4.3 Receiver Interfaces 198
5.4.4 Outbound Processing 199

5.5 The XI Message Protocol 203

5.6 Configuration Scenario 204
5.6.1 Creating a Configuration Scenario from Scratch (Manually) 204
5.6.2 Creating a Configuration Scenario from a Model 205

5.7 Value Mapping 206
5.7.1 If/Else Logic 206
5.7.2 Fixed Values 207
5.7.3 Value Mapping 208

5.8 Business-to-Business Integration 209
5.8.1 Business-to-Business On Premise 209
5.8.2 Business-to-Business Managed Services 210
5.8.3 Trading Partner Management 210

5.9 Axis Framework 212

5.10 Representational State Transfer Adapter 215

5.11 Message Alerting 218
5.11.1 Alert Rule Overview 219
5.11.2 Creating an Alert Rule 222
5.11.3 Editing or Deleting a Rule 225

5.12 Publish the Service in the Services Registry 225

5.13 Integration Directory Programming Interface (Directory API) 226

5.14 Exercise: Working with the Integration Directory 229
5.14.1 Exercise Description 229
5.14.2 Exercise Solution Approach 230
5.14.3 Exercise Step-by-Step Solution 230

5.15 Summary 238

6 Building an Integration Flow 241

6.1 SAP NetWeaver Developer Studio 242
6.1.1 Installing SAP NetWeaver Developer Studio 242
6.1.2 Setting Up SAP NetWeaver Developer Studio 243
6.1.3 Enterprise Integration Patterns and User-Defined Templates 246

6.2 Basics of Creating and Configuring an Integration Flow 248
6.2.1 Creating an iFlow 248
6.2.2 Configuring an iFlow 250
6.3 iFlow Example 256
6.3.1 Creating Products, Software Components, Business Systems, and Technical Systems in the SLD 256
6.3.2 Importing SLD Objects into the ES Repository in SAP NetWeaver Developer Studio 259
6.3.3 Create Enterprise Service Repository Objects 260
6.3.4 Create Directory Objects: Import Business Systems and Create iFlows 268
6.3.5 Testing the iFlow Scenario 272
6.3.6 Monitoring the Scenario 272
6.4 New Features of the Process Integration Designer 273
6.4.1 Export Objects from the Integration Designer 273
6.4.2 Automatically Deploy after Import 274
6.4.3 Version History and Deployment Status 276
6.5 Supporting Multiple Senders for Your iFlow 277
6.6 Exercise: Building an Integration Flow 280
6.6.1 Exercise Description 280
6.6.2 Exercise Solution Approach 281
6.6.3 Exercise Step-by-Step Solution 281
6.7 Summary 289

7 Administration and Monitoring in AEX 291

7.1 Administration 291
7.1.1 Central Administration Tool 292
7.1.2 SAP NetWeaver Administrator 293
7.1.3 SAP NetWeaver Application Server Java 294
7.1.4 SAP Process Integration Monitoring (pimon) 294
7.1.5 SAP Management Console 294

7.1.6 Config Tool 295
7.1.7 Administration Using Telnet 296
7.2 Monitoring 296
7.2.1 SAP NetWeaver Administrator 296
7.2.2 SAP Process Integration Local Monitoring 297
7.2.3 SAP PI Central Monitoring with SAP Solution Manager 300
7.2.4 Message Retention 303
7.2.5 User-Defined Message Search 303
7.3 Troubleshooting 309
7.3.1 Configuring Log and Traces 311
7.3.2 Using the Log Viewer 313
7.4 Summary 316

8 Migrating Interfaces from SAP PI Dual Stack to SAP PO 317

8.1 Migration Strategies 317
8.2 Migrating System Landscape Directory Content 321
8.2.1 Products 321
8.2.2 Software Components 322
8.2.3 Technical System 323
8.2.4 Business System 323
8.3 Migrating Enterprise Services Repository Content 324
8.3.1 Exporting Objects 324
8.3.2 Importing Objects 325
8.4 Migrating Integration Directory Content 325
8.4.1 Manually 326
8.4.2 Using the Migration Tool 327
8.5 Summary 337

PART III Business Process Management and Composition

9 Introduction to SAP BPM and BPMN 2.0 341

9.1 Managing Business Processes 342

9.2 SAP Business Process Management 344
9.2.1 SAP BPM versus SAP Business Workflow 345
9.2.2 BPM before SAP BPM 347
9.2.3 SAP BPM Main Components 349

9.3 Business Process Model and Notation 2.0 351
9.3.1 Swimlanes 353
9.3.2 Artifacts 356
9.3.3 Flow Objects 358
9.3.4 Connections 367

9.4 Summary 369

10 Creating Your First SAP BPM Process 371

10.1 SAP BPM Positioning and Development Environment 371
10.1.1 Positioning 372
10.1.2 Setting Up Your Development Environment 373

10.2 Creating and Modeling an SAP BPM Process 377
10.2.1 Demonstration Scenario 377
10.2.2 Building an SAP BPM Process: Overview 379
10.2.3 SAP NetWeaver Developer Studio Perspective Concept 380
10.2.4 Create a Project in SAP NetWeaver Developer Studio 382
10.2.5 Creating a Process for Your BPMN 387
10.2.6 Creating a BPMN Model 390

10.3 Configuring the BPMN Model 396
10.3.1 Data Objects 396
10.3.2 Creating Data Structures 397
10.3.3 Importing XSD and WSDL 399

10.3.4 Process Pool Properties 399
10.3.5 BPMN Flow Objects 406

10.4 Flow Objects 407
10.4.1 Events 407
10.4.2 Tasks 418
10.4.3 Activities 428
10.4.4 Gateways 447
10.4.5 Artifacts 450

10.5 Build and Deploy Your Process 452
10.5.1 Steps for Building a Process 453
10.5.2 Steps for Deploying a Process 454

10.6 Advanced Mapping 455
10.6.1 Mappings 455
10.6.2 Options in Mapping Assignment 457
10.6.3 Automatic Mapping 458
10.6.4 Custom Functions 460

10.7 Implementing Error Handling 462

10.8 Combining SAP BPM and the AEX 464
10.8.1 Message from SAP BPM to the AEX 466
10.8.2 Message from the AEX to SAP BPM 467
10.8.3 Leverage an ES Repository Mapping in SAP BPM 468

10.9 Exercise: Creating an SAP Business Process Management Process 470
10.9.1 Exercise Description 471
10.9.2 Exercise Solution Approach 471
10.9.3 Exercise Step-by-Step Solution 472

10.10 Summary 498

11 Applying Advanced SAP BPM Concepts and Extensions 501

11.1 Service-Oriented Architecture Configuration 501
11.1.1 Configuration for an Automated Activity 502
11.1.2 Configuration for a Start Event or Intermediary Event 515

11.2 Testing and Running an SAP BPM Process ... 517
11.2.1 Process Repository Overview ... 517
11.2.2 Process Testing ... 519

11.3 Custom Enterprise Java Bean Functions ... 523
11.3.1 Create EJB and EAR Development Components ... 523
11.3.2 Create the Enterprise Java Bean ... 526
11.3.3 Build and Deploy ... 529
11.3.4 Create a New Enterprise Java Bean Function ... 530

11.4 Using the Claim Check Pattern ... 532
11.4.1 Create Interfaces ... 534
11.4.2 Create Mappings ... 534
11.4.3 Configure the Channel ... 534
11.4.4 Retrieve the Large Message from SAP BPM ... 535
11.4.5 Update the Status of the Large Message from SAP BPM ... 536

11.5 SAP BPM Application Programming Interface ... 536
11.5.1 Prerequisite to Using the SAP BPM API ... 537
11.5.2 Implementation Aspects and Examples ... 537

11.6 SAP Business Process Management OData ... 539
11.6.1 OData Services for Tasks and Task Data ... 541
11.6.2 Error Handling ... 541

11.7 Using the Push API to Access SAP BPM Lifecycle Events ... 542
11.7.1 Accessing Events through a Message Driven Bean ... 543
11.7.2 Accessing Events through a Java Message Service API ... 544

11.8 Debugging and Troubleshooting SAP BPM Processes ... 545
11.8.1 Place Breakpoints in the Process ... 546
11.8.2 Add a Debug Configuration ... 547

11.9 Tuning SAP BPM-Related Performance Parameters ... 549

11.10 Best Practices for Your SAP BPM Application ... 551
11.10.1 BPMN, Mapping, and Parallelism ... 551
11.10.2 Task Related ... 552
11.10.3 Gateways ... 552
11.10.4 Looping ... 553
11.10.5 Data Object ... 553
11.10.6 Correlation ... 553

11.10.7 Error Handling ... 554
11.10.8 Housekeeping ... 554

11.11 Exercise: Applying Advanced SAP BPM Concepts and Extensions ... 554
11.11.1 Exercise Solution Approach ... 555
11.11.2 Exercise Step-by-Step Solution ... 555

11.12 Summary ... 566

12 Combining SAP BPM and UI Technologies 567

12.1 Web Dynpro Java User Interface Technology ... 567
12.1.1 Generating a Web Dynpro User Interface ... 569
12.1.2 Post-Configuration Steps ... 574

12.2 Integrating SAPUI5 into an SAP BPM Process ... 578
12.2.1 SAPUI5 Technology Platform ... 579
12.2.2 Model-View-Controller Concept ... 580
12.2.3 SAPUI5 Components ... 581
12.2.4 Other SAPUI5 Concepts ... 583
12.2.5 Integration Steps ... 583

12.3 Other User Interface Technologies ... 592
12.3.1 Visual Composer ... 592
12.3.2 Adobe Offline Forms ... 592
12.3.3 Support for Custom User Interface Technologies ... 593

12.4 Summary ... 595

13 SAP Business Rules Management 597

13.1 How Business Rules Work ... 597

13.2 SAP Business Rules Management ... 601
13.2.1 Rules Composer ... 603
13.2.2 Rules Manager ... 605
13.2.3 Rules Engine ... 609

13.3 Modeling Business Rules with Rules Composer 610
13.3.1 Create the Rules Composer Development Component 610
13.3.2 Adding Context to the Rules 612
13.3.3 Creating a Ruleset 614
13.3.4 Flow Ruleset 620
13.4 Testing Business Rules 623
13.5 Best Practices for Modeling Business Rules 625
13.5.1 Separate Decision Logic from Other Types of Logic 625
13.5.2 Reuse and Extend before Building 626
13.6 Exercise: SAP Business Rules Management 627
13.6.1 Exercise Solution Approach 628
13.6.2 Exercise Step-by-Step Solution 629
13.7 Summary 650

14 Implementing Java Proxies 653

14.1 Java Proxy Concept and Considerations 654
14.2 Implementation Approaches 655
14.2.1 Outside-In Approach 655
14.2.2 Inside-Out Approach 656
14.3 Technical Implementation 656
14.3.1 Development Environment 656
14.3.2 Developing a Server Java Proxy 659
14.3.3 Developing a Client Java Proxy 675
14.4 Building an Orchestration 687
14.5 Exercise: Implementing Java Proxies 688
14.5.1 Exercise Solution Approach 689
14.5.2 Exercise Step-by-Step Solution 689
14.6 Summary 714

15 Administration and Monitoring Message Processing in SAP BPM 715

15.1 Monitoring 715
15.1.1 SAP Business Process Management System Overview 716
15.1.2 Process Repository 717
15.1.3 Process Management 718
15.1.4 Task Management 723
15.1.5 SAP BPM Inbox 725
15.1.6 Business Logs 727
15.1.7 SAP BPM Action Monitor 728
15.1.8 Process Troubleshooting 730
15.1.9 Rules Business Logs 730
15.1.10 SAP BPM Analytics Dashboard 731
15.2 Administration 733
15.2.1 Process Data Archiving 733
15.2.2 Log Viewer 734
15.3 Summary 736

16 Migrating ccBPM from SAP PI to SAP PO 737

16.1 Motivation for Migration 737
16.2 Migration Approach 740
16.2.1 Analyze the As-Is Integration Processes 743
16.2.2 Translate and Redesign 745
16.2.3 Export and Reuse Enterprise Services Repository Objects 748
16.2.4 Migrate and Adapt Configuration Scenarios 751
16.3 Recommendations 755
16.4 Summary 757

PART IV Advanced Concepts

17 SAP Cloud Platform Integration for SAP PO 761

17.1 Enable Cloud Integration Content in SAP PO 762

17.2 Reusing Cloud Integration Content 763
17.2.1 Download Cloud Integration Content 764
17.2.2 Deploy the Cloud Integration Content 766

17.3 Monitoring 767
17.3.1 Monitoring the Integration Gateway Component 767
17.3.2 Monitoring Messages Related to the Deployment of Cloud Integration Content 769

17.4 Summary 770

18 Additional Components for SAP Process Orchestration 771

18.1 Component Model 772
18.1.1 Product 773
18.1.2 Software Components 773
18.1.3 Development Component 774
18.1.4 Dependencies among Development Components 776
18.1.5 Public Parts 779

18.2 SAP NetWeaver Development Infrastructure 781
18.2.1 Change Management Services 783
18.2.2 Design Time Repository 798
18.2.3 Component Build Service 801

18.3 SAP Composite Application Framework 804
18.3.1 Design Time Aspects 805
18.3.2 Runtime Aspects 811

18.4 Service Registry 812

18.5 Enhanced Change and Transport System 815
18.5.1 SAP PI-Related Transports 816
18.5.2 Transports for Non-SAP PI Java Objects 819

18.6 Exercise: Create an SWCV 821
18.6.1 Exercise Solution Approach 821
18.6.2 Exercise Step-by-Step Solution 822

18.7 Summary 830

19 Landscape Setup Considerations 833

19.1 Java System Configuration 833
19.1.1 Java Sizing and Setup Considerations 834
19.1.2 Java System Architecture 841
19.1.3 Java Central Services 842
19.1.4 Java Parameter Tuning 843

19.2 Handling Certificates 846
19.2.1 Certificate Key Storage 846
19.2.2 Encryption of Message Content on Database Level 846

19.3 Housekeeping 847
19.3.1 Archiving 847
19.3.2 Deletion 847
19.3.3 Restarting 847
19.3.4 Recovery 848

19.4 Monitoring 848
19.4.1 Runtime Workbench 849
19.4.2 Wily Enterprise Manager 850
19.4.3 SAP Management Console 851
19.4.4 SAP Solution Manager Monitoring 853
19.4.5 Tracing 854
19.4.6 JVMMON 856

19.5 Summary 857

Appendices

A **Orchestration Outlook** 859

B **The Authors** 885

Index 887

Index

A

ABAP
- *business system* ... 97
- *data supplier* ... 107

Abstract
- *interface* ... 751
- *service interface* ... 743

Acceptance ... 502
Access Control List ... 776
Active Directory ... 62
Activities ... 362
- *automated activity* ... 435
- *type* ... 362

Actor ... 739
Adapter
- *file/FTP(S)* ... 189
- *HTTP_AAE* ... 189
- *IDoc_AAE* ... 190
- *JDBC* ... 189
- *JMS* ... 189
- *mail* ... 189
- *marketplace* ... 189
- *RFC* ... 190
- *SOAP* ... 189
- *specific attributes* ... 182
- *type* ... 189
- *types, support header mapping in AEX* ... 200

Adapter engine ... 180
Adapter module
- *asynchronous processing* ... 182
- *DynamicConfigurationBean* ... 182
- *JNDI* ... 186
- *PayloadSwapBean* ... 181
- *PayloadZipBean* ... 181
- *StrictXml2PlainBean* ... 181
- *TextCodepageConversionBean* ... 182
- *XiHeaderValidationBean* ... 182
- *XMLAnonymizerBean* ... 181

Adapter object ... 152
Add complex type ... 398
Add mapping ... 270
Administration ... 715
Administrator via principals ... 401
- *assign roles* ... 402

Adobe offline forms ... 592
- *SAP BPM integration* ... 593

Advanced Adapter Engine (AAE) ... 34, 54
Advanced Adapter Engine Extended (AEX) ... 33, 291, 349, 372, 495, 502
- *administrative tools* ... 292
- *diagnose integration scenarios* ... 293
- *expose server proxy service* ... 675
- *integration challenges* ... 39
- *message monitoring* ... 298
- *message monitoring error* ... 313
- *messaging exchange* ... 291
- *monitoring* ... 296
- *SAP PI monitoring with SAP Solution Manager* ... 301

Agency identifier ... 174
Alert notification ... 301
Alert rule ... 173, 218
- *configuration* ... 222
- *create* ... 222
- *creation via Integration Directory* ... 224
- *creation via SAP NetWeaver Administrator* ... 222
- *default rule* ... 221
- *editing/deleting* ... 225
- *information* ... 219
- *message header rules* ... 223
- *message statuses* ... 223

Annotation ... 357
Application and scenario communication ... 227, 513, 521, 811
Application communication ... 493, 504
Application controllers ... 581
Application programming interface (API) ... 71, 226, 227, 859
- *communication channel* ... 229

Application service ... 810
Application Thread Manager ... 51

Application thread utilization 851
Applications 38, 52
Archiving data 554
Archiving tasks
required roles 733
Arguments 548
Artifact 356, 450
annotation 357, 452
data object 357, 396, 450
AS Java engine 50
ASMA 182, 186
activate on communication channel 187
Assign system
sender and receiver 269
Assigned users 201
Association connector 358
Asynchronous communication 140
mapping 145
Asynchronous process
SAP BPM 653
Attribute 162
Auto assign 508
Automapping
advanced 459
options 458
select 459
simple cases or name based 459
Automated activity 362
add 435
boundary event 437
configure 437
input mapping 437
interface 436
looping 436
output mapping 437
Automated rule-based decisions 601
Automatic mapping 458
matcher 458
Automation of monitoring 853
Availability and Performance
monitoring 716
Availability and Performance
work center 80
Axis framework 212
verify installation 213
AXIS Servlet 214
Axis-based channel setup 214

B

B2B
cloud 210
integration 209, 876
Integration Cockpit 210
mapping functions 210
mapping template 210
number range object 210
on premise 209
rapid-deployment solution 210
scenarios 71
Trading Partner Management 210
Backend 738
Backend application 94
expose functionalities 45
BE (Best Effort) 141
Best practices 551
correlation 553
data object 553
error handling 554
gateway 552
housekeeping 554
looping 553
Boundary event 360, 415, 416, 435
configuration 416
critical exception 433
noncritical exception 433
BPEL4WS 738
definition 739
BPM 42
Breakpoint 546
BRFplus 602
BRM system 43
Build directory 803
Buildspace, broken DC details 804
Business activity monitoring (BAM) 78, 342
Business and IT alignment 351
Business application design 568
Business component 173, 175
create 176
maintain server interface 176
maintain service interface 176

Business content 61
Business decisions 44
Business entity 805
Business log 554, 608
Business Logs tool 727, 730
Business object
application service 809
create 806, 808
createMultiple 809
delete 809
deleteMultiple 809
findAll 809
findByMultipleParamaters 809
implementation 808
main structure 806
modeler 806
modeler operations 807
read 809
update 809
updateMultiple 809
Business process
ccBPM 54
conditions and rules 598
diagram (BPD) 351
engine 36
model and document tool 351
Business process model
lifecycle 342
suite 343
versus workflow 345
Business Process Model and Notation
(BPMN) 341, 351, 371, 551, 722, 738, 739
add human activity 391
artifacts 356
create model 390
data object 357
data types 396
diagram 488
event triggers 395
flow element 352
flow objects 406
flow objects, properties 406
functions 396
model 484, 501
model, configuration 396
process 395
Business Process Model and Notation
(BPMN) (Cont.)
process, create 387
reporting data sources 395
rule sets 395
service interfaces 396
tasks 395
Business rule 598
access control management 606
add context 612
automated 45, 599
creation approach 604
deploy and execute 609
embedded 604
execution logs 607
generate web service 624
methods 598
model 610
modeled as rule flow 603
modeling best practices 625
standalone service 604
test 623
types and use 601
Business rules engine, lack of 54
Business rules management 43
Business Rules Management System 600
Business system 97, 173
attributes 99
create 98
export objects from SAP PI SLD 323
groups 99, 100
import 268
import from SLD 175, 268
select role 98

C

Cache Manager 51
Calculation 601
Calculator service 667
CallSapAdapter 181
CBS 783, 801
buildspace 803
central compilation 802
Request Receiver 802
web UI 802

CBS (Cont.)
web UI, access ... 803
Central adapter engine
monitoring ... 296
Central Administration tool ... 292
Central database ... 50
Central repository of rules ... 44
Central SAP PI monitoring ... 301
Central Services Registry ... 813
Central Services Server (SCS) ... 842
Central Technical Configuration (CTC) wizard ... 762
Centralized system management ... 294
Certificate ... 846
key storage ... 846
Channel adapter ... 269
Claim check pattern ... 532
approach ... 533
configure the channel ... 534
create interfaces ... 534
create mappings ... 534
Class Loader Manager ... 52
Client Java proxy ... 675
Client proxy ... 711
generate ... 679
wrapper class ... 681
Cloud integration content ... 763
deployment ... 766
Cloud-to-cloud integration ... 862
Cluster Manager ... 51
Cluster node ... 549
CM Services ... 783
CTS system ... 784
deployable export ... 820
integrated with CTS+ ... 786
modes ... 819
source export ... 820
standalone ... 784
synchronize ... 792
Collaboration profile ... 173
communication component without party ... 173
party ... 173
Collection pattern ... 471
Combining SAP BPM and AEX ... 464
Common Information Model (CIM) ... 106
instances and classes ... 103, 106
send data to SLD ... 107
Common language ... 351
Communication channel ... 177, 231
adapter type ... 179
direction ... 179
file name ... 270
link to communication component ... 179
message protocol ... 180
module ... 181
module processor ... 181
Monitor ... 849
monitoring ... 854
parameters ... 179
polling interval ... 179
receiver channel ... 179
sender channel ... 179
setting for WS connectivity with SAP BPM ... 504
source directory ... 270
transport protocol ... 180
Communication component ... 174
business component ... 175
business system ... 175
without party ... 186
Communication profile ... 515
assign ... 516
create ... 516
Communications ... 39
Comparators ... 619
Component Build Service
see CBS ... 783
Component model ... 772
create software component ... 773
development component ... 774
possible DC types ... 775
product ... 773
software component ... 773
Component monitoring ... 853
Composite application ... 805
Composite Designer perspective ... 389
Composite Explorer ... 382, 394
Composite service ... 47, 810
create ... 687
Composition layer ... 46

Condition evaluation ... 448
Conditions ... 618
Config Tool ... 295
Configuration ... 82
Configuration and monitoring ... 77
Configuration management ... 517, 520
Configuration Manager ... 52
Configuration scenario ... 204
create from model ... 205
create manually ... 204
remove objects ... 205
Configure the deployed service ... 671
Connecting object ... 367
association ... 368
data flow ... 368
message flow ... 368
sequence flow ... 368
Connectivity adapter ... 45
Consolidated installation ... 57
Consolidated monitoring ... 57
Consumer proxies ... 813
Consumer service groups ... 511, 514
Core Components Technical Specification (CCTS) ... 149
Correlation condition ... 412
Cost savings ... 46
CPA cache monitor ... 298
Cross-component Business Process Management (ccBPM) ... 34, 55, 347, 737
abstract interface ... 138
as is analysis ... 743
integration processes, migration strategy ... 744
migration steps ... 742
projects ... 756
translate integration process ... 745
vs. SAP BPM ... 740
CTS server destination ... 819
CTS+ ... 749
configuration ... 816
configure CTS parameters ... 818
create a destination ... 816
export SLD object ... 114
test export functionality ... 123
transports ... 123
Custom EJB function ... 523
Custom function
create ... 460
EJB function ... 460
expression and mapping ... 460
Custom Java application ... 52

D

Data forwarding ... 122
Data object ... 357, 396, 569
add to process ... 450
analyze ... 722
binding ... 573
characteristics ... 357
Data structure, create ... 397
Data supplier ... 107
ABAP-based system ... 107
change in Java system ... 108
configuration ... 121
destination ... 108
forward to SLD ... 122
Java-based systems ... 108
send data ... 107
Data transformation ... 441
Data type ... 149, 159
core/aggregated ... 149
freestyle ... 149
XSD support ... 396
Database ... 50
Database clustering ... 840
Database Manager ... 51
DC
activate and deactivate ... 518
add software components ... 787
auto-deployment ... 785
create ... 523, 571, 775
create dependencies ... 571
create from scratch ... 790
dependency ... 776
dependency, creating ... 776
export and import settings ... 784
import from development configuration file ... 790
import options ... 789
importing from SLD ... 789
management ... 784

DC (Cont.)
name ... 790
public part ... 779
repository and build tool location ... 785
runtime system ... 784
SAP BRM ... 610
settings ... 784
SLD ... 784
software component definition ... 785
source ... 775
synchronize ... 792
transport settings ... 785, 819
Debug
add configuration ... 547
breakpoint ... 546
configuration ... 547
exception circumstances ... 546
prerequisites ... 546
start ... 548
Debugging ... 545
Debugging function ... 546
step into ... 546
step out ... 546
step over ... 546
Decentralized adapter engine ... 180
Decision ... 601
Decision table ... 622, 628
Decoupling ... 56
Default gate ... 447
Default matcher ... 458
DEFAULT_ALERTRULE ... 221
Defensive checks ... 553
DeltaV ... 799
Dependency ... 101
conditions ... 776
synchronize ... 778
Deploy time dependency ... 777
Deployment descriptor ... 658
Design object ... 141, 151
add structure ... 142
Design time dependency ... 777
Design Time Repository ... 783, 798
detection of version conflicts ... 800
features ... 799
inactive workspace ... 803
Design Time Repository (Cont.)
managing multiple versions ... 799
server side ... 799
Design Time Repository Browser ... 799
Developer trace access ... 735
Developer traces ... 314
Development ... 76
Development component ... 105, 502, 571, 774
create ... 678
Development configuration ... 783
Development environment setup ... 373
Development infrastructure ... 523, 657, 756
Development objects ... 775
Diagram editor designer ... 810
Dictionary ... 805
Directory API ... 327
Distributed Management Task Force (DMTF) ... 106
DNS round-robin load balancing ... 837
Dual stack ... 737
migration ... 318
migration procedure ... 740

E

EAR DC ... 526
build and deploy ... 685
build and deploy to server ... 669
create ... 523
Eclipse ... 87
configure ... 865
EDI message exchange ... 210
EJB ... 554
build ... 529
create ... 526
custom function ... 564
deploy ... 529
function, create ... 530
module ... 805
EJB DC ... 526, 657
create ... 523, 660, 678
Embedded subprocess activity ... 438
Embedded subprocess, aspects ... 438
Encryption on database ... 846

End event ... 360
event trigger ... 409
input mapping ... 410
intermediary event ... 410
Enqueue replication server ... 838
Enqueue server ... 49
Enqueue server process ... 842
Enterprise application ... 661
Enterprise Application Archive DC ... 658
Enterprise application DC create ... 661
Enterprise integration pattern ... 246, 248
claim check pattern ... 532
dynamic conditions ... 247
message translator ... 247
point-to-point ... 247
recipient list ... 247
Enterprise Java Bean (EJB) ... 523, 654, 700
Enterprise Service Browser ... 657
import SLD objects ... 260
Enterprise service bus (ESB) ... 36, 291, 349
integration challenges ... 39
integration with ... 40
Enterprise Services Registry (ES Registry) ... 67, 134, 225
services in SOA landscape ... 135
Enterprise Services Repository ... 45, 347
Enterprise Services Repository (ES Repository) ... 61, 133
exercise ... 153
folder ... 142
functional blocks ... 134
import software component metadata ... 136
namespaces best practices ... 143
namespaces, predefined ... 144
Entity services ... 804
EO (Exactly Once) ... 141
EOIO (Exactly Once in Order) ... 141
Error and exception handling ... 601
Error handling ... 541
possible solutions ... 463
ES Builder ... 63, 134, 158
mapping object ... 146
recommended system requirements ... 64
tasks performed from ... 65
ES Registry ... 67
publish ... 225
ES Repository ... 61, 473
activate objects ... 267
content migration ... 324
export objects ... 324
export or reuse objects ... 748
import Objects ... 325
import the service interface ... 663
import WSDL ... 397
inbound service interface ... 659
Web UI ... 69
ES Repository content ... 61
ES Workplace ... 61
Escalation event ... 361
Events ... 359
end event ... 409
start event ... 408
Exception events
CompletionDeadlineIsCritical-Exception ... 434
CompletionDeadlineIsNonCritical-Exception ... 434
HandleExceptionErrorEvent ... 434
StartDeadlineIsCriticalException ... 434
StartDeadlineIsNonCriticalException ... 434
Exception handling ... 750
Exception type ... 433
Exchange of data ... 38, 41
Exchange of messages ... 42, 174
Exclusive gateway ... 392
Exercises ... 124, 153, 229, 280, 470, 554, 627, 688, 821
Export Service link ... 795
Extensions ... 554
External definitions ... 149

F

Facade ... 52
Fault message type ... 150
Filters ... 314
Firing queue ... 617
Fixed value mapping ... 207
Flow object ... 358, 407
activity ... 362

Flow object (Cont.)
boundary event ... 360
connect ... 367
end event ... 360
event ... 359
events ... 407
gateway ... 364
intermediate event ... 360
start event ... 360
Flow ruleset ... 620
Format lanes ... 407
Framework API ... 52
Freestyle data type ... 149
FTP integration ... 157
FTP server ... 233
Full automatic synchronization
bidirectional ... 116
unidirectional ... 116

G

Galaxy Core Service ... 550
Gateway ... 364, 447, 644
abstract or undefined ... 447
event-based choice ... 366, 448
exclusive choice ... 365, 447
parallel join ... 366, 450
parallel split ... 366, 448
split ... 552
types ... 365, 447
uncontrolled merge ... 367, 450
Global configuration ... 193

H

High availability
database ... 840
web dispatcher setup ... 836
with standalone enqueue ... 838
History Service access ... 798
Host ... 540
Housekeeping ... 554, 847
archiving ... 847
deletion ... 847
recovery ... 848
restarting ... 847
HTML5 ... 579
HTTP error status code ... 541
HTTP methods ... 540
DELETE ... 540
GET ... 540
POST ... 540
PUT ... 540
HTTP protocol ... 540
Human activity ... 362
add boundary event ... 434
boundary event ... 433
general ... 428
input mapping ... 431
looping ... 429
output mapping ... 432
task ... 430
Human-centric ... 740
processes ... 346

I

IDE ... 603
Identifier ... 180
IDoc ... 327
IDoc monitoring ... 298
iFlow ... 87, 173, 241, 280, 689, 764
activate ... 254
add and remove receivers ... 253
add mapping ... 253
apply channel template ... 252
assign a service interface ... 250
assign referenced channel ... 252
assign sender and receiver ... 250
business component ... 250
configure ... 250
configure channel attributes ... 251
configure communication channel ... 251
consistency check ... 254
create ... 248, 285
deploy ... 254
deployment log ... 254
deployment trace ... 254
iFlow ID ... 248
import business system ... 250
monitor message processing in runtime ... 272

iFlow (Cont.)
test ... 272, 288
iFlows ... 86, 860
If-then rule ... 614, 622
Import the Service Interface ... 692
Inbound processing ... 192, 193
Inbound service interface ... 663
Incident management ... 301
Industry solutions ... 137
Inside-out approach ... 656
Installation options ... 53
Instance memory threshold ... 716
Integrated configuration (ICO) ... 70, 173, 192, 233, 752
advanced settings ... 201
configuration troubleshooting ... 194
Integration Advisor ... 876, 879
interface specifications ... 878
mapping guideline ... 878
runtime ... 878
testing ... 878
Integration Builder ... 70, 154
configuration and monitoring tools ... 291
launch ... 171
Integration Directory ... 130, 147, 171, 229, 239
automate migration ... 327
component ... 173
content migration ... 325
create business component ... 176
create configuration scenario ... 204
editor area ... 172
manual migration ... 326
menu ... 172
migrate existing content ... 751
migration classic configuration ... 326
migration ICOs ... 326
migration integration process ... 326
object types ... 172
using the Migration tool ... 327
Integration engine ... 54
Integration gateway component ... 762
Integration gateway component monitor ... 768
Integration pattern ... 138
stateful ... 138
stateless ... 138
Integration pattern (Cont.)
Tentative Update and Confirm or Compensate ... 139
Integration patterns
see Enterprise integration patterns ... 246
Integration platform-as-a-service (iPaaS) ... 860
Integration process ... 173
Integration scenario create ... 133
Integration visibility configuration ... 78
Interface ... 52, 53
building process ... 171
channel monitoring ... 854
determination ... 192
Interface Directory
Swing UI tool ... 255
Interface object ... 151
Intermediary event
automated activity ... 502
configuration ... 515
event trigger ... 410
Intermediate event ... 360
Intermediate message event ... 411
delivery ... 412
matching ... 411
Intermediate timer event ... 413
expression ... 413
time offset ... 413
Internationalization and localization ... 601
Internet Communication Manager (ICM) ... 48, 49, 844
Investigator ... 850

J

JAR files ... 53
Java
name reservation ... 104
Java Archive (JAR) ... 657
Java Archiving Cockpit ... 733
access ... 733
Java Central Services ... 842
instance ... 835
Java client proxy ... 689
Java enterprise runtime ... 50, 51
Java instance ... 48

Java Message Service (JMS) 45
Java Naming and Directory Interface
(JNDI) 185
browser 186
Java only 737
Java parameter
database connection pool configuration 845
FCAServerThreadCount 843
garbage collection configuration 845
ICM timeout values 844
tuning 843
Java proxy 653
client 675
compile 669
deploy 669
develop 656
generate Java Bean skeleton 665
generate wrapper 682
implementation 653
implementation approaches 655
orchestration 687
orchestration, create 687
runtime 654
Java Proxy Runtime Monitor 849
Java server proxy 700
Java system
architecture 841
configuration 833
Java Web Start 65
Java Web Start application 171, 230
Java-only installation 36
Java-only stack 36
Java-only system
setup considerations 834
sizing 834
JAX-WS 654
JNDI
access 530
filter Java Bean 530
object name 530
jQuery 579
JSON 579
JVMMON 856

K

Key performance indicators (KPIs) 346, 850

L

Landscapes
create 96
functionalities 94
Lane 355
Large message 534
Legacy application 45
Library 52, 53
Licensing Manager 52
Load balancer
advantages 839
Demilitarized Zone (DMZ) 839
Load balancing 839
Local or central ES Repository 62
Local system 505
LocalDevelopment 660
Lock and cache administration 292
Lock collision 49
Lock request 49
Lock table 49
Locking Manager 51
Log and traces 311
Log ID 541
Log level setting 554
Log Manager 51
Log shipping 840
Log Viewer 84, 313, 541, 734
access 734
show log 508
Logging 201
Login credentials 548
Looping 553
configuration 429
currentCollectionItem 430
numberOfCompletedIterations 430
Parallel For Each 436
Parallel For Each option 429

M

Manage processes 849
Manage tasks 849
Mapping 144, 487, 551
leverage mapping in SAP BPM 468
reuse in SAP BPM 468
Mapping activity 441
general 442
looping 442
mapping 442
mapping function 441
mapping line 441
Mapping as a service (MaaS) 468
Mapping assignment 457
append 458
merge 458
set 457
Mapping complexity 552
Mapping object 146
MAX-RESTART 847
Memory consumption 552
Message
alerting 218
alerting, process steps 219
from SAP BPM to the AEX 466
mapping 146
Monitor 848
monitoring 272
protocol 509
protocol, WS 503
retention 303
server 49
trigger 416
type 150
Message mapping editor 166
Message monitoring 769
Message structure 475
Message trigger 491
Message type 159, 161, 166
MessageStatusUpdater 536
MESSAGE_ID_LIST 536
PROCESS_INSTANCE_ID 536
MessageStatusUpdaterBean 536
Metadata 61, 805
Metering of service calls 194
Migration 737, 742
approach 740
big bang/phase-based 744
reasons 737
recommendations 755
translate and redesign 745
Migration strategy 317
in-place upgrade 318
side-by-side migration 318
Migration tool 78, 752
access 328
add new system 335
channel migration 328, 333
configuration 328, 335
conflict resolution 333
considerations 327
create change list 332
create rule 336
features 328
mass migration 333
migrate communication channel 333
renaming rule 336
required roles 329
scenario migration 328, 329
scenario selection 330
steps overview 333
unsupported objects 337
Mobile 47
Modeling 151, 739
Modeling environment 61
Model-view-controller 579, 580
controller 580
model 580
view 580
Monitor Java heap utilization 851
Monitor system 851
Monitoring 86, 715, 767, 848
Monitoring Home screen 77
Monitoring tools
authorizations and roles 715
Move-corresponding 552
Multiple application servers 839
Multithreaded 48

N

Name reservation 104
Namespace 142
Namespace prefix 104
Network communication overhead 56
Node 841
None intermediary trigger 415
Notification activity 392, 445
  general 445
  mail 446
  specify receiver 445
  to 445
NotificationTask
  configuration 589
NotificationTask, configuration 589
NWA_READONLY 716
NWA_SUPERADMIN 716

O

Object Management Group 351
Object, export or import 321
Object-oriented modeling 106
OData 579
  API 539
  wrapper 539
OData services 882
  for task data 541
  for tasks 541
Online rules editing tool 606
Openness 539
Operation mapping 144, 165, 168
Operations 82
Operations for SAP Cloud Platform
  Integration 865
Orchestration 687
OriginalMessageAccessor 535
Outbound processing 192, 199, 236
Outbound service interface 162, 283
Outside-in approach 655

P

Parallel processing 552
Parallel split gateway 552
Parallel web dispatcher
  setup 837
Parallel web dispatcher, setup 836
Parallelism 551, 552
Parameterized subject 552
Parameterized text, create 426
Parameters file
  create 871
  populate 871
Party 173
  Dun & Bradstreet Corporation 174
  *EAN - International Article Numbering
    Association* 174
  SCHEME 174
path_segment 540
Payload 548
Performance 742
Perspective 380
  Debug 546, 548
  Java EE 526
  Process Development 547
  *SAP Process Integration
    Administration* 244
  *SAP Process Integration
    Designer* 244, 247, 248
  SAP Process Integration Runtime 244
pimon 77, 294
  AEX monitoring tools 297
  SAP NetWeaver Administrator 296
Point-to-point channel 269
Point-to-point integration 39
Polling interval 550
Pool
  administrator 402
  assign administrator 402
  assign administrator via expression 403
  characteristics 354
  configure 399
  object 353
Pool Manager 51
Port 540
  number 48
Potential owner 431
Presentation layer 47
Principal 401
Principal expression
  getPrincipal 404

Principal expression (Cont.)
  getPrincipalByUniqueName 404
  getPrincipals 405
Principal, rules and functions 405
Process 547
  all running processes 719
  error and failed processes 719
  instance ID 720
  perform administrative actions 723
  see details 720
  Task Management tool 723
  view definition flow 721
  view process history 722
Process data
  archiving 733
  volume 554
Process definition 721
Process driven organization 348
Process flow
  alternative path 360
  exceptional situations 361
  execute different rules 620
  pool 353
Process instance context data 722
Process instance details access 719
Process integration
  action 148
  scenario 147
  tools 59, 64
Process Management tool 718
  access 718
  tasks 718
Process modeling and execution 351
Process pool properties 399
  description 400
  enable principle propagation 400
  expression 401
  Make this pool active 400
  name 400
  parameterized texts 401
  principals 400, 401
  variables 401
Process Repository 454, 493, 520, 717
  access 517, 717
  list DCs 518
  overview 517
Process task attributes 724
Process Troubleshooting tool 730
  access 730
ProcessDefinition 538
ProcessDefinitionManager 538
Processes and tasks 520
ProcessStartEvent 538
Product, create 101
Production 502
Program sldreg 110
Protocols 45
Provider system 505
Proxy 190
  generation 137
Public part 776, 779
  assembly 780
  compilation 780
  create 780
  manage entities 780
  purposes 779
Publishing services 225

Q

Quality of service (QoS) 141, 270

R

RAR files 53
Receiver 192, 195
  enhanced receiver determinations 196
  not operation-specific 196
  operation-specific 196
  routing condition 195
  simple receiver and conditions 195
Receiver agreement 173, 192
Receiver channel 287
Receiver communication channel
  configuration 270
Receiver component 163
Receiver determination 192, 197
Receiver interface 192, 198, 235
Redesigning business processes 755
Referenced subprocess activity 439
  boundary events 441
  input mapping 440

Referenced subprocess activity (Cont.)
looping ... 440
output mapping ... 440
process ... 440
Referenced subprocess aspects ... 439
Related integration server ... 98
Reliability ... 47, 56
Remote Function Call (RFC) ... 83, 502
Renaming rule ... 337, 754
Report activity ... 442, 443
REST ... 539
REST adapter ... 215
features ... 217
REST web services ... 882
REST wrapper ... 536
Restart ... 549
RESTART-BLACKLISTED ... 848
RESTful ... 859
RESTful web services ... 215
RfcAFBean ... 181
Round-robin ... 837
Routing ... 41, 601
Rule ... 614
script ... 621
Rule action
assign ... 616
categories ... 616
decision table ... 618
evaluate decision table ... 617
execute ... 616
reevaluate ... 616
retract ... 617
Rule flow ... 621
configure ... 642
Rules and functions
Boolean ... 456
Durations, Dates and Times ... 456
Generic ... 456
Numeric ... 456
Principal ... 456
String ... 456
Rules Business Logs tool ... 730
Rules Composer ... 603, 610, 620, 627, 642
Rules Composer DC
import existing XSD file ... 613
Rules engine ... 600, 609
authorization and administration ... 606
reexamine all rules ... 616
Rules Manager ... 605, 620
editing tool ... 606
Rules repository ... 600
Rules simulation ... 600
Runtime Workbench (RWB) ... 78, 849

S

SampleTimer ... 437
SAP API Management ... 815, 859
SAP B2B TPM
functional profile ... 211
trading partner agreement ... 211
trading partner profile ... 211
SAP backend ... 504
application ... 45
SAP BPM Action Monitor ... 728
access ... 728
SAP BPM activities ... 428
activity ... 428
human activity ... 428
SAP BPM API ... 538
BPEM-FACADE ... 537
classes ... 537, 539
dependencies ... 537
ENGFACADE ... 537
features ... 536
main objects ... 537
package ... 539
prerequisites ... 537
ProcessDefinitionManager ... 537
ProcessInstanceManager ... 538
ProcessModelManager ... 538
ProcessStartManager ... 538
roles ... 537
TaskInstanceManager ... 538
SAP BPM Inbox ... 589
access ... 726
configuration ... 725
process ... 725
required roles ... 727
task activities ... 725
SAP BPM Java API ... 583
SAP BPM Mail Server Connectivity
configuration ... 446
SAP BPM OData ... 539
base URL ... 540
extra options ... 540
principles ... 540
SAP BPM Open Data protocol
see SAP BPM OData ... 539
SAP BPM process
advanced mapping ... 455
boundary event ... 463
build ... 453
create mapping ... 455
data mapping ... 455
debugging ... 545
deploy ... 454
download resources ... 519
error end event ... 462
implementing error handling ... 462
input data ... 521
instance, steps to start ... 519
lanes ... 388
name ... 388
new pool ... 388
obtain WSDL ... 522
orientation ... 388
performance parameters ... 549
project ... 388
rules and functions ... 456
running ... 517
start ... 519
start externally ... 522
start internally ... 520
steps ... 387
termination event ... 463
test ... 519
testing ... 517
troubleshooting ... 545
tuning ... 549
SAP BPM queue
content ... 729
settings ... 729
summary ... 729
SAP BRM ... 601
SAP Business Process Management
(SAP BPM) ... 27, 33, 344, 350, 501, 653, 715
action ... 549
Analytics Dashboard ... 731, 732
API ... 536
application ... 551
business process visibility ... 344
call web service steps ... 466
Composite Designer perspective ... 380
create product ... 382
create software component ... 383
custom functions ... 460
custom UI integration steps ... 594
DC ... 546
developer ... 545
fundamental concepts ... 341
gateway ... 447
human tasks ... 592
implementation steps ... 379
integrate SAPUI5 ... 584
integration of custom UI technologies ... 593
interaction with the AEX ... 465
landscape scope ... 348
main components ... 349
monitor process instances ... 718
object ... 538
OData ... 583
positioning ... 371
process ... 371
Process Development perspective ... 380
process implementation ... 377
Process Modeling perspective ... 381
RFC ... 502
supported scenarios ... 349
system overview ... 716
system overview access ... 716
system overview capacity ... 716
troubleshooting ... 733
vs. ccBPM ... 740
WS ... 502
XI 3.0 ... 502
SAP Business Rules Management
(SAP BRM) ... 27, 33, 350, 597, 730, 741
components ... 602
design time ... 603
rule implementation ... 44

SAP Business Workflow 345, 738
versus BPM 345
SAP BusinessObjects Business Intelligence (SAP BusinessObjects BI) 346
SAP Central Services 49
SAP Central Services instance 49
components 49
SAP Cloud Platform Integration 761, 859, 860, 876
application edition 862
certificates and security artifacts 865
cloud 862
cloud solutions 860
communication channels, configure 873
connectivity options 865
designer 865
development guide 864
features 863
flows 764
iFlow, create 867
license 862
mapping, create 872
Message Mapping 872
monitor using Eclipse 875
monitoring 874
monitoring using the WebUI interface 875
multitenant architecture 863
on-premise 860, 861
pattern template 867
plugin 864
professional package 863
project, deploy 874
repository connection 866
sender and receiver systems 870
standalone integration platform edition 862
tenant 762
tenant account 865
user-defined template 869
WebUI 861
workbench and tools 864
SAP Cloud Platform Workflow 882
SAP Code Exchange 536
SAP Composite Application Framework (SAP CAF) 46, 771
business object 805
SAP Composite Application Framework (SAP CAF) (Cont.)
capabilities 804
configuration 811
core project 805
entity services 807
runtime aspects 811
SAP Composition Environment 34, 35, 73
SAP CR content 119
SAP Enterprise Portal 47
SAP Exchange Infrastructure (SAP XI) ... 34, 133
2.0 34
3.0 34
SAP Gateway 859
SAP Industry adapter
CIDX 191
RosettaNet 191
SAP industry adapters 191
SAP JVM 842
SAP landscape 38
SAP license 52
SAP Management Console (SAP MC) 851
features 852
SAP NetWeaver 7.3 EHP 1 SP 09 539
SAP NetWeaver Administrator 79, 293, 496, 504, 715, 826
configure deployed service 685
create alert rule 222
Log Viewer 313
SAP NetWeaver Administrator (NWA) 60
SAP NetWeaver AS 750
SAP NetWeaver AS ABAP 34
SAP NetWeaver AS Java 34, 93, 294, 657
advanced offline configuration 295
applications 50
cluster 47
communication 49
configure data suppliers 120
console administration 296
deploy business rules 609
synchronization 49
system architecture 47
system components 52, 53
system components or services 50
SAP NetWeaver AS Java cluster 49
installation 49

SAP NetWeaver Developer Studio 60, 86, 87, 241, 242, 285, 352, 472, 555, 629
business rules 603
configure for Java proxy 657
create business object 806
create data type 262
create Enterprise Service Repository objects 260
create message mapping 265, 266
create message type 263
create namespace 261
create project 382
create Rules Composer DC 611
create service interface 264
create XSD 397
development components overview 387
download link 373
Enterprise Service Browser 245
Enterprise Service Repository connection 245, 376
installation 242
perspective 89
perspective concept 380
PI Explorer 248
PI Tools 243
preferences 375
Rules Composer 603
SAP NetWeaver AS Java 375
settings 374
setup 243
switch perspective 381
versions 374
SAP NetWeaver development infrastructure 103, 104, 741, 771, 781
access 782
access export service 795
activate name server role 104
activities 800
advantages 781
archive file 794
CBS 801
check in 799
check out 799
client side 799
Export Service 794
SAP NetWeaver development infrastructure (Cont.)
Export UI 820
History Service 798
Import Service 798
inbox path 793
include source/archives 796
resolve conflict 800
runtime systems 792
stop on error 796
support package and patch levels 796
synchronisation recommendation action 793
SAP NetWeaver foundation 52
SAP PI 349
7.1 34
7.11 34
7.3 34
dual stack 53
dual stack migration to SAP PO 317
dual to single stack 318
local monitoring 297
monitoring 294
separate installations 54
tools 243
SAP PI Explorer 268
SAP PI Tools Configuration 244
SAP PO 33
application server 319
business process support 320
calling services 767
capabilities 47
components info 374
custom table 320
data and value mapping 320
determine version 374
development tools and IDE 319
foundation 47
history 36
home page 59
integration content 88
interface configuration 320
mapping 320
migration option 318
positioning 45

SAP PO (Cont.)
server ... 546
single-stack architecture ... 351
SAP Process Integration → SAP PI
SAP Process Orchestration → SAP PO
SAP Quick Sizer tool ... 834
SAP Service Marketplace ... 61
SAP SuccessFactors ... 860
SAP Web Dispatcher ... 835
SAP_BPM_Navigation ... 716
SAP_BPM_ODATA ... 540
SAP_BPM_ODATA_INBOX ... 540
SAP_BPM_SuperAdmin ... 716
SAP_BPM_SuperDisplay ... 716
SAP_BPM_TRIGGER_EVENT ... 537
SAPUI5
application ... 583
complete task ... 591
component ... 581
component file ... 582
component structure ... 582
composite control ... 583
Eclipse plugin ... 579
faceless component ... 582
generate a DC ... 585
integrate into SAP BPM ... 578
integrate with SAP BPM ... 584
libraries ... 579
notepad control ... 583
SAP BPM Inbox ... 725
steps to generate UI ... 585
UI component ... 582
UI generation ... 567
UI library ... 583
Scalability ... 47
SCPMappingBean ... 534
mappingid ... 535
persistuntil ... 535
xpath ... 535
Security ... 42
Security artifacts ... 767
Sender agreement ... 173, 192
Sender authorization ... 869
Sender communication component ... 510
Sender component ... 160
configure in SAP NetWeaver Administrator ... 510
Server process ... 48, 49
Server proxy ... 659
expose service ... 675
generate ... 665
generation wizard ... 665
test ... 674
Service ... 53
Service composition ... 46
Service group ... 85
configure ... 513
create ... 469, 506
Service group and reference, configuration ... 504
Service interface ... 137, 159
add ... 176
import ... 678
integration pattern ... 138
message type and data type ... 149
Service orchestration workflow ... 737
Service reference ... 502
configuration ... 510
configuration steps ... 511
WS ... 503
Service Registry ... 771
access ... 812
activate ... 813
capacities ... 812
Service_name ... 540
Service-oriented architecture (SOA) ... 27, 35, 40, 63, 133
configuration ... 493
maturity level ... 135
strategy ... 40
Services ... 52, 550
bundling ... 46
Session Manager ... 51
Shipment unit ... 142
Simple Object Access Protocol (SOAP) ... 45, 843
Simplify communications ... 41
Single point of failure (SPOF) ... 835
Single Service Administration ... 521

SLD software component model ... 102
one software component model ... 103
three software component model ... 103
two software component model ... 103
SMTP ... 48
SOA configuration ... 501
intermediary event ... 502
RFC ... 512
start event ... 502
WS ... 503
XI 3.0 ... 509
SOA work center ... 84
SOAP client ... 237
SOAP UI client tool ... 289
SoapUI ... 569
Social media platform ... 140
Software catalog ... 75, 100
product ... 101
software component ... 101
Software component ... 101, 124, 125
export content ... 794
specify release ... 384
specify support package ... 384
Software component version
see SWCV ... 142
Software unit ... 76
Source
namespace and rank ... 117
SSL certificate ... 839
Staging ... 201
Standalone EJB module ... 53
Standalone proxy ... 656
Standalone resource adapter module ... 53
Standalone web module ... 53
Standard services ... 45
Standards ... 45
Standby database ... 840
Start event ... 360
configuration ... 515
event trigger ... 408
output mapping ... 409
start condition ... 409
Start process ... 520
Stateful message processing ... 738
Stateless ... 139
Step over ... 549
Subprocess ... 363
Swap space ... 842
SWCV ... 76, 154
create ... 257
main types ... 136
namespaces ... 142
Swimlanes ... 353
Swing client ... 241
SXMB_ADM ... 77
SXMB_MONI ... 77
Synchronize Service ... 791, 792
Synchronous communication ... 140
mapping ... 145
System connections ... 516
System landscape
setup ... 833
System Landscape Directory
product ... 773
registration ... 850
System Landscape Directory (SLD) ... 73, 93, 134, 745
add data or content to ... 112
administration ... 110
automatic bridge forwarding ... 115
bridge ... 107
catalogs ... 119
clear data cache ... 67
connect to CTS+ ... 122
create business system ... 259
create product ... 256
create software component versions ... 257
create technical system ... 258
data supplier ... 108
data supplier configuration ... 109
export business systems ... 323
export data to target ... 122
export products from SAP PI ... 321
export software components ... 322
export technical systems ... 323
full automatic synchronization ... 116
full sync (active) ... 118
functional areas ... 74
group ... 100
import and export ... 112
import business systems ... 323
import objects into ES Repository ... 259

System Landscape Directory (SLD) (Cont.)
import products into SAP PO 322
import software components 322
import technical systems 323
information categories 94
landscape 74
manual export and import 114
migrate content 321
naming convention 118
product repository 100
registering systems 107
self-registration 120
server information 111
server logs 111
status 111
stop and start 111
strategies 113
synchronization 113
System-centric process 346, 743
Systems 38
group logically 96
logical name 97

T

Task 418
assign UI 573
best practices 552
claim 725
complete 541
create 422
filter 725
forward 725
mapping with process context 577
open 725
operations 541
potential owners 576
read data 541
release 725
Task attributes 418
activation time 422
administrators 421
completion deadline 422
component 419
custom actions 420
custom attributes 420
Task attributes (Cont.)
documentation 419
excluded owners 420
expiration time 422
name 419
notification 419
parameterized texts 421
priority 420
sharing 419
start deadline 422
variables 421
Task Management tool
access 724
features 723
TaskInput 431
Tasks service
URIs list 541
Technical monitoring
overview 853
Technical system 129, 175
business system 97
maintenance 94
manually create 95
Telnet 296
Test 502
Third party developed adapter
Seeburger 191
Third-party applications 539
Third-party developed adapter 191
Informatica 191
iWay 191
Thread 851
utilization 851
Tightly coupled applications 39
Toggle breakpoint 546
Total cost of ownership (TCO) 36
TPZ 749
Tracing 854
HTTP and SOAP communication channels 854
XPI Inspector 855
Transaction
RZ70 108
Transformation 41
Transport 124
activity 820

Transport (Cont.)
approach 815
mechanism 321
non-SAP PI Java objects 819
path 815
SAP PI-related 816
Troubleshoot 545
Troubleshooting 84, 309
Tuning 549
Two-phase commit protocol 139

U

UDDI 812
UI technology
choose 385
Composite Application Framework (CAF) 386
dictionary 386
JSP/Servlet 386
process modeling 386
Rules Composer 386
service composition 386
Visual Composer 386
Web Dynpro 386
UME 403
user 445
UME actions 540
Unique identifier 538
Universal Discovery, Description, and Integration (UDDI) 62
Universal worklist
alternative 725
URI 538
URI concept 538
Usage dependency 101
User interface (UI) 567
generation 569
User Management 294
User roles 762
User-defined function (UDF) 187
User-defined message search 294
User-friendly interface 45

V

Validation 601
Value mapping 206, 208
fixed values 206
group 173
manual approach 209
replication from an external source 209
Virtual business system
label 175
Virtual logical container 142
Virus scan 193
Virus scanning
by adapter 193
Visual Composer 350, 592
Visual containers 353
Vulnerable security connection 40

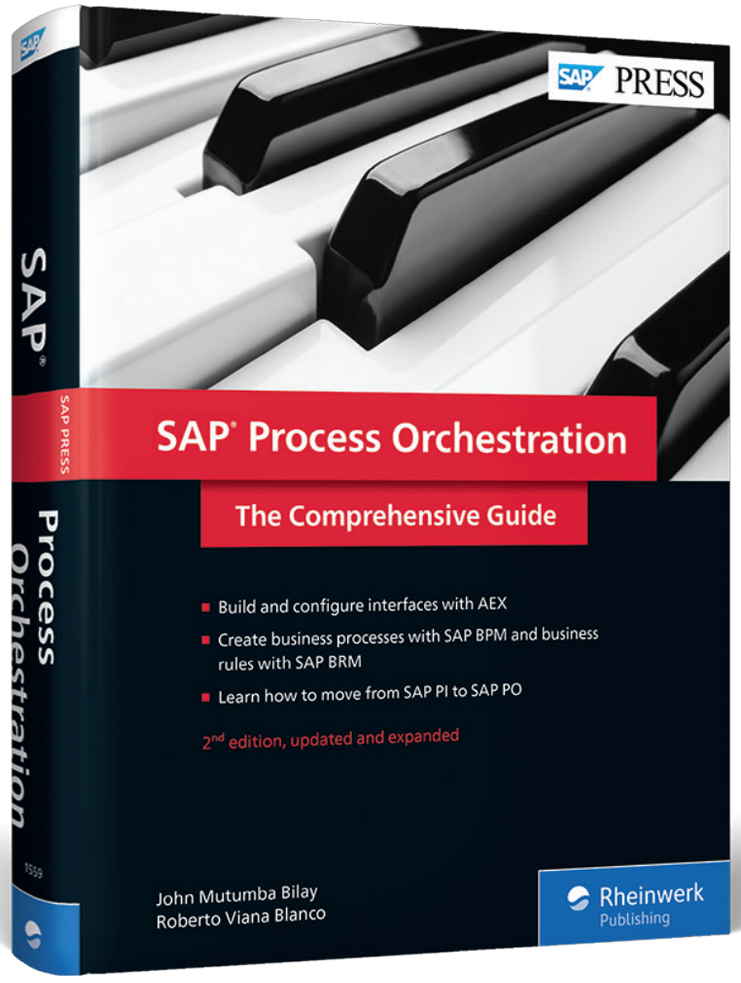
W

WAR files 53
Web Dynpro
ABAP 568
application 605
assign task principals 575
generate UI 569
Java 568
post-configuration steps 574
UI generation steps 569
Web service
authentication 673
import external WSDL 677
message security 673
settings 671
transport protocol 672
Web Service Description Language (WSDL) 226, 281, 474, 604
details 237
generate client proxy from 679
import steps 399
Web Services Navigator 294, 497, 521
WebDAV 799
WebserviceX 282
WebUI 69, 861

Wily Enterprise Manager 850, 851
performance analysis 850
Work center 79
Workflow engine 345
WS connection, assign communication profile 516
WS-Security 214

X

XI 3.0 502
XI 3.0 message protocol 466
scenarios 509
XI message protocol 179, 203
differences from SOAP standard 203
XIMailAdapterBean 181
XISOAPAdapterBean 181
XML 548
XML DAS 733
XML Data Package (XDP) 593
XML file 649
XML Schema Definition (XSD) 396
create 397
import 399
XPI Inspector 855



John Bilay, Roberto Viana Blanco

SAP Process Orchestration: The Comprehensive Guide

908 Pages, 2017, $79.95
ISBN 978-1-4932-1559-1

www.sap-press.com/4431

John Mutumba Bilay is a senior software engineer and enterprise integration consultant at Rojo Consultancy. With more than 14 years of international experience in information technology, including eight specifically devoted to SAP, his speciality is software development, integration, and business process automation. You can visit his company's website at http://www.rojoconsultancy.com/en/.

Roberto Viana Blanco is an experienced SAP integration architect and co-founder of Rojo Consultancy. He has more than 17 years of SAP integration experience working across a number of industries. He has been working with SAP since the early versions of SAP NetWeaver Exchange Infrastructure (XI) and has helped several organizations adopt SAP Process Integration and then SAP Process Orchestration.

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