Applies to:
SAP NetWeaver Composition Environment 7.1, SAP NetWeaver Process Integration 7.1.

Summary
This tutorial consists of two major parts. First, it gives an overview of the typical components within a SOA landscape with special focus on SOA middleware components. Beside a generic, SAP-independent description of the typical SOA components, a mapping to the corresponding SAP products is given. The second part describes the various installation and configuration steps to set-up a SOA landscape that consists of SAP components only, i.e. an SAP NetWeaver PI 7.1, an SAP NetWeaver CE 7.1 and an SAP ERP back-end. The goal is to provide the reader with a complete overview of the required and optional configuration steps in an SAP SOA landscape. For the sake of completeness, this tutorial will briefly discuss the additional non-SAP components that are needed for the connection between the SAP PI and a 3rd party Enter Service Bus (ESB), in this particular case the IBM WebSphere ESB v6.1 and OpenSSL.

This tutorial ignores the aspects of comprehensive SOA monitoring and identity management due to the focus on the co-existence of Enterprise Service Buses (ESBs).

Please note that this tutorial is part of the documentation about heterogeneous SOA landscapes. To find out more please go to the Collaboration Workspace or have a look at this blog.

Author: Peter Kulka
Company: SAP AG
Created on: 25 November 2008

Author Bio
Peter Kulka received his PhD in Computer Science / Computer Graphics from the University of Auckland, New Zealand. After teaching Java / C++ programming courses at the University of Auckland, Peter joined SAP AG in 1999. Until November 2005 Peter was responsible for the product definition of the SAP NetWeaver Application Server, SAP’s Java EE 5 compliant application server. Currently Peter is a Solution Architect in SAP’s Global Ecosystem and Partner Group, where he advises SAP partners on how to architect their Java solutions based on SAP NetWeaver. Peter presented at leading industry conferences such as JavaOne, SAP TechEd, and OOP.
# TABLE OF CONTENTS

**Overview of SOA Landscapes**
- Service Provider ................................................................. 3
- Service Registry ........................................................................... 3
- Detailed Documentation of Services / SOA Methodology ............. 3
- Service Repository ........................................................................ 4
- Enterprise Service Bus ............................................................. 4
- Service Consumer ........................................................................ 6

**Required Configuration Steps of an SAP SOA Landscape**
- Import service interfaces, models, GDTs .................................. 8
- Implement Enhancement Package ............................................ 12
- Publish ES in the SAP Services Registry ................................. 12
  - ABAP Back-End Systems 7.0 < Support Package 14 ................. 12
- Check Services Registry proxy (NW PI) .................................... 14
- Configure Services Registry proxy (NW CE) ............................. 17
- Configure Visual Composer (optional) ........................................ 18

**Set-up of non-SAP SOA Middleware Components** .................. 18
Overview of SOA Landscapes

In this tutorial I am going to discuss the typical components of a SOA landscape and provide an overview of the necessary configuration steps.

Service Provider
Let’s start with components of a SOA landscape. First of all you need at least one back-end system, e.g. an SAP ERP or SAP CRM system that provides certain business functionality through services. For the SAP Business Suite, services are provided through enhancement packages that need to be installed on the corresponding back-end system. These services are called Enterprise Services (ES).

Service Registry
For the consumption of services, a service consuming application needs two types of information:

- The service interface description and
- the corresponding back-end system, where the implementation of a service is located or deployed.

This information is usually available from a service registry through so-called Web Service Description Language (WSDL) files. WSDL is an open standard and WSDL files contain service definitions and optionally service endpoints. A service endpoint contains a Uniform Resource Locator (URL), which specifies the location, i.e. the back-end system, of the deployed service. Every Web service consumption framework can import WSDL files and typically generate from the information of the WSDL file all artifacts that are required for the corresponding service calls. For more information on how to use the Services Registry provided by SAP refer to SAP Online Help (Discovering Services in the Services Registry).

There is an open standard for service registries called Universal Description, Discovery and Integration (UDDI), which makes the interoperability of service registries possible. SAP’s Services Registry (SR) contains a UDDI compliant service registry plus SAP specific components such as Classification and Free text search to navigate through the services more quickly.

Detailed Documentation of Services / SOA Methodology
Beside the technical aspects of service calls, it is also important to fully understand the business semantics of the services and the service interfaces. The developer or a business process expert, who wants to consume a service, needs to know the semantics behind the pure technical information provided by the WSDL file, such as a detailed description of the services, interfaces and data types, related business objects and business processes, etc. For Enterprise Services
provided by the SAP Business Suite, this information together with tons of very detailed documentation can be obtained from the Enterprise Service Workplace (ES Workplace) on SDN: https://www.sdn.sap.com/irj/sdn/esworkplace.

**Service Repository**

In order to provide new services or enhance existing services, you need to create new service interfaces or modify existing ones. In both scenarios, you need to access the service interface definitions and their meta data information, such as the used data types, business objects or models. This information is usually stored in a service repository. The Enterprise Services Repository (ESR) is SAP’s service repository (SAP Online Help and SDN: https://www.sdn.sap.com/irj/sdn/soa-modeling). It is worth mentioning that SAP defined a harmonized set of data types for all Enterprise Services of the entire SAP Business Suite based on open standards. These data types are called Global Data Types (GDT). For more information on GDTs, please have a look at the SAP Online Help, at this blog or at this blog (which also includes a link to the GDT catalogue).

Please note that the term Enterprise Services Repository denotes both the Enterprise Services Repository and the Services Registry. Some vendors do not distinguish between a service registry and repository in their product offerings, e.g. IBM's Web Service Registry and Repository (WSRR).

**Enterprise Service Bus**

While a service registry is required for a SOA landscape, the usage of an Enterprise Service Bus (ESB) is optional. Unfortunately there is no exact definition on what an ESB really is:

“There is some disagreement on whether an enterprise service bus is an architectural style, a software product, or a group of software products. While use of an ESB certainly implies adherence to a particular architecture, the term “enterprise service bus” is almost always used to denote the software infrastructure that enables such an architecture.” (Wikipedia).

Even though the definition above is pretty generic, the following capabilities are generally expected from an ESB infrastructure:

- **Service invocation** – the ability to call services. This requires the support of synchronous and asynchronous transport protocols as well as the possibility to locate and bind services.
- **Routing** – distribute messages according to certain rules, e.g. dependent on the content of a message (content-based routing) or according to specific predefined rules (rules-based routing).
- **Mediation** – reconcile incompatible protocols, data formats and interaction patterns of disparate connected applications and systems (e.g. though adapters, protocol transformation, service interface mapping).
• Messaging – provide an infrastructure to ensure that messages are passed and exchanged between applications and systems in a reliable way (this includes message processing and message transformation).
• Quality of Service – important capabilities for productive usage of an ESB (e.g. security, reliability or transaction handling).
• Management – manage service calls between consumers and providers with regards to monitoring, administration, audit and logging.

The capabilities listed below are sometimes also considered as ESB capabilities:

• Business Process Management (BPM) – can range from simple split, merge and/or correlation of messages up to complex process orchestration with end user interaction and workflow integration. It is quite common to provide separate BPM-engines that are specialized for this purpose. However, most ESBs provide at least some rudimentary BPM capabilities.
• Event processing – interpretation and identification of events as well as event correlation, i.e. making sense of incoming events.

The figure below shows a logical view on an Enterprise Service Bus. The SAP NetWeaver Process Integration (SAP NetWeaver PI) is SAP’s ESB product. The SAP NetWeaver PI provides all the capabilities mentioned above. It furthermore provides the Cross-Component Business Process Management (ccBPM) component to model and execute business processes. For more sophisticated BPM tasks, SAP ships the SAP NetWeaver Business Process Management (SAP NetWeaver BPM) component with the SAP NetWeaver Composition Environment (SAP NetWeaver CE).

Besides these technical capabilities the SAP NetWeaver PI also provides business content for a tight semantic integration of the applications within the SAP Business Suite.
Having a service registry, a service repository and optionally an ESB in place, it is possible to consume services provided by the back-end systems. There are many ways to consume services, which depend on the chosen programming language, service consumption framework and underlying development platform. Typical platforms are Java EE compliant application servers or Microsoft’s .NET platform. A Java developer for instance, may use Axis to consume services. It is also possible to consume services in ABAP or Ruby. SAP NetWeaver CE provides many service consumption tools in addition to a Java EE compliant application server, such as the SAP NetWeaver BPM for BPM tasks, the Visual Composer to model UIs for composite applications and the Composite Application Framework (CAF) to develop more complex composite applications, which require back-end abstractions, service adaptation capabilities or local persistence.

Service registries, service repositories and ESBs are typically regarded as SOA Middleware. However, back-end systems and composition environments are usually not considered as being part of the SOA Middleware.
Required Configuration Steps of an SAP SOA Landscape

This section provides an overview of the required installation and configuration steps to set-up a SOA landscape that consists of SAP products.

Since we aim to build a heterogeneous SOA landscape with multiple ESBs, we installed the SAP NetWeaver PI 7.1 beside the SAP NetWeaver CE 7.1 and a Discovery System (https://www.sdn.sap.com/irj/sdn/soa-discovery), which serves as an ERP back-end. More information about the installation of the SAP NW CE 7.1 can be found in this blog and information about the installation of the SAP NW PI 7.1 can be found here.

The SAP NetWeaver CE and the SAP NetWeaver PI can both host a Services Registry and an Enterprise Services Repository. Since we only need one registry and one repository, we will only use the ESR of the SAP NetWeaver PI, which is SAP’s recommendation in this case (SAP Online Help). This means that we will configure the SAP NetWeaver CE, so that it will use the SAP Services Registry and Enterprise Services Repository of the SAP NetWeaver PI. The figure below lists the corresponding configuration steps, which will be outlined in the following sections.

![Configuration Steps of an SAP SOA Landscape](image)

Steps:
1. Import service interfaces, models, GDTs
2. Implement Enhancement Package
3. Publish ES in the Service Registry
4. Check SR proxy (NW PI)
5. Configure SR proxy (NW CE)
6. Configure Visual Composer

©SAP AG 2008
**Import service interfaces, models, GDTs**

After the installations of the SAP NetWeaver PI and the SAP NetWeaver CE have finished successfully, the first step is to load the service interfaces, the models and GDT into the Enterprise Services Repository of the SAP NetWeaver PI, since the Enterprise Services Repository does not contain any business content after the installation. As noted above, this step is technically only required if you plan to provide your own services or if you want to enhance or modify existing service definitions. However, importing the business content into the ESR makes the discovery of Enterprise Services easier, since it is possible to use the high level models to drill down to the services of choice. Furthermore, the content can also be used in the ES Browser within the ABAP Workbench of a connected ABAP backend system to navigate to the appropriate implementation.

Before you can import business content into the Enterprise Services Repository, you need to select the appropriate content packages. The figure below shows the granularity of shipment units.
The two units **SAPGLOBAL 2.0** and **SAPGLOBAL MODEL 2.0** contain the definitions of the GDTs. These two units can be used for all SAP products. The models for the Enterprise Services are shipped through the **ESM Integration 1.0** unit. This package can be used for the entire SAP Business Suite. Additional models are available for specific product versions, such as **ESM ERP 603** for ERP or **ESM SCM 5.1** for SCM. The service definitions and their implementations are available for the various software component versions.

The software component version for a particular Enterprise Service operation can be found in the Services Registry or on the ES Workplace, when looking up an Enterprise Service. For example, to look-up the software component version of the Enterprise Service operation “Find Customer Address Basic Data by Name and Address” in the ES Workplace ([https://www.sdn.sap.com/irj/sdn/esworkplace](https://www.sdn.sap.com/irj/sdn/esworkplace)), navigate to “ERP Process Components” -> “Business Partner Data Management” -> “Query Customer In” -> “Find Customer Address Basic Data by Name and Address”. The software component version is listed in the technical data section, here “ESA ECC-SE 603”.

You should at least select the two packages that contain the GDTs, the two packages for the service models and all packages for the software component versions that contain the Enterprise Service operations you need.

SAP customers can download the selected SAP NetWeaver PI content units from the SAP Marketplace. To download the corresponding units of the Enhancement Package 3 for ERP, for example, go to [http://service.sap.com/swdc](http://service.sap.com/swdc) and navigate to “Downloads” -> “SAP Support Packages” -> “Entry by Application Group” -> “SAP Application Components” -> “SAP ERP” -> “SAP ERP ENHANCE PACKAGE” -> “SAP ENH PACK 3 FOR SAP ERP 6.0” -> “Entry by Component” -> “ESR 7.10”. After adding the content packages to your download basket, it may look like this:
At the time this document was written, SAP released the Enhancement Package 4. To download the corresponding units for the Enhancement Package 4 for ERP go to http://service.sap.com/swdc and navigate to “Downloads” -> “SAP Support Packages” -> “Entry by Application Group” -> “SAP Application Components” -> “SAP ERP” -> “SAP ERP ENHANCE PACKAGE” -> “EHP4 FOR SAP ERP 6.0” -> “Entry by Component” -> “ESR content”. Here you find the Global Data Types and the ES Models (for the SAP Business Suite and for SAP ERP).

When you have downloaded the corresponding content files from the SAP Marketplace, you need to unzip and import the files into the Enterprise Services Repository. To import the content files, go to the Enterprise Services Repository start page (http://<hostname>:<port>/rep) and choose “Enterprise Services Builder”. After you have successfully logged on to the Enterprise Services Repository, you can import the unzipped content files via the menu “Tools” -> “Import Design Objects” as shown below:
In the pop-up window choose “Client” and select the .tpz file that you want to import in the file selection box. Finally confirm your choice by pressing the “Import” button:

After a successful import of the business content, you can access the new content in the Enterprise Services Repository (the example below shows the imported Global Data Types):
Please have a look at this blog for more details on how to import PI content. Information on how to import and use PI content can also be found here.

**Implement Enhancement Package**

In order to Service-enable SAP ERP, you need to install the units for the latest SAP Enhancement Package (EHP) that contain the Enterprise Services you want to consume. The following page on the SAP Service Marketplace contains all relevant information about the SAP Enhancement Packages and how to install them: [http://service.sap.com/erp-ehp](http://service.sap.com/erp-ehp). This blog explains how to install SAP Enhancement Pack 4 on SAP ERP 6.0.

**Publish ES in the SAP Services Registry**

Assuming that you have successfully installed and deployed the required Enhancement Packages that contain all Enterprise Services that you need, the next step is to publish the services in the Services Registry. This procedure differs depending on release and support package level of the underlying ABAP stack in the back-end.

**ABAP Back-End Systems 7.0 < Support Package 14**

For an AS ABAP 7.0 with a support package <= 13, this procedure is described in detail in the document "CE, ESR, AND DISCOVERY SYSTEM – BUILDING AN END-TO-END ENTERPRISE SOA SCENARIO" (paragraph 3.3, page 10), which can be found on here. Additionally this blog describes this procedure as well.
Here is just a rough sketch of the process:

- Logon to you SAP back-end system (ABAP stack) and
- create a new RFC destination that points to the Services Registry using the transaction “SM59”. You need to provide a name and description for the destination and you must know the hostname (parameter “Target Host”) and port number (parameter “Service No.”) of the Services Registry. The “Connection Type” should be “G” and the “Path Prefix” should be “/ESRegistryWS/BasicAuthConfig?wsdl&style=document&mode=standard”. Save and test the connection by pressing the “Connection Test” button.
- Use transaction “WSPARAM” to create an Services Registry parameter,
- use transaction “LPCONFIG” to create a logical port pointing to the Services Registry and
- use transaction “WSPUBLISH” to finally publish the ES.

**ABAP Back-End Systems 7.0 > Support Package 13**

For an AS ABAP 7.0 with a support package level >= 14, the tools to configure (WSPARAM, LPCONFIG, WSCONFIG), to administer (WSADMIN) and to publish (WSPUBLISH) Web services were replaced by one single tool, the SOA Manager.

Since the publication of ES using the SOA Manager is described in detail in the SAP Online Help, we just outline the main process steps here:

- Logon to you SAP back-end system (ABAP stack) and
- start the SOA Manager using the transaction “soamanager”. Here it is possible to create a logical port that points towards the Services Registry. This step is necessary to provide a technical connection between the ABAP back-end system and the Services Registry. This procedure is described in detail in the SAP Online Help. Alternatively this step is also described here.
- Now it is necessary to configure or customize the Services Registry, so that the connection, which was defined in the previous step, can now be used to publish the ES to the Services Registry. More information about this step can be found on the bottom of this page.
• Now you can start to publish ES from the back-end system to the Services Registry using the SOA Manager. There are two options to publish ES: automatically or explicitly (SAP Online Help).

In most cases you will probably publish ES automatically (SAP Online Help). Here the Service definitions are published together with their endpoints in the Services Registry. Whenever service endpoints are created, updated, or deleted in the back-end system, the information about the service endpoints is automatically published to the Services Registry in accordance with the active publishing rules.

Furthermore it is possible to publish multiple Services at the same time. This mass publication of Services can be configured in the SOA Manager (SAP Online Help). To publish multiple Service at once you can define configuration profiles (“Technical Configuration” -> “Profile Management”) that can later be assigned to a set of ES in a configuration scenario. A configuration profile defines the transport and security settings that are required to call an ES. For example, you can define a configuration profile for ES that are consumed only within your enterprise and another configuration profile for ES that are public available. In the mass configuration menu (“Business Administration” -> “Mass Configuration”), you can assign a configuration profile to a set of ES, so that the consumption of the ES is constrained by the corresponding settings of the assigned configuration profile. The assignment of ES to configuration profiles triggers automatically the publication of the ES to the default Services Registry.

For the explicit publication of ES select the ES and the Services Registry, where the selected ES should be published, and start the publication manually. Publishing ES in this way is more suitable for single services. More information about can be found in the SAP Online Help.

Please note that there is a limitation with the publication of ES from an ABAP back-end system to the SAP Services Registry: If the ABAP back-end is part of an SAP NetWeaver PI, which means that it is a dual-stack installation, then it is not possible to publish ES from that client (Mandant) of the ABAP stack that is coupled to the Java stack. By default this is typically client 001.

Check Services Registry proxy (NW PI)
When the publication of the ES from the SAP ERP back-end to the SAP Services Registry has finished, it is possible to browse and test the ES in the SAP Services Registry. The SAP Services Registry can be started from the homepage of the SAP NetWeaver Application Server or accessed directly through the link http://<hostname>:<port>/sr. To check, whether the ES are published correctly search for one service that you published before. In our example we are going to check the ES “CustomerERPAddressBasicDataByNameAndAddressQueryResponse_In” by searching for “CustomerERP*”. If the service is listed, select it and go to the tab “Endpoints”. 

©SAP AG 2008
The Web Service Navigator should now start up and asked you for a user and password for the corresponding back-end system. Now you can test the Enterprise Service. More information about browsing the Services Registry can be found in the SAP Online Help or in this blog.

If you do not see the published ES in the SAP Services Registry, there are basically three things to double-check: the two destinations “UDDI_DESTINATION” and “CLASSIFICATION_DESTINATION” and the proxy definition for the SAP Services Registry, if defined.
The configuration of the two destinations is described in detail in the SAP Online Help. Furthermore, this blog demonstrates in a screen cam how to set-up the two destinations. Make sure that the hostname, port, system name, URL, user and password entries are correct.

The proxy definition for the SAP Services Registry can be accessed from the SAP NetWeaver Administrator (http://<hostname>:<port>/nwa) -> “SOA Management” -> “Business Administration” -> “Web Service Administration”. In the field “Search By:” select “Proxy Definition Name” and search for “Service*” as the name of the proxy is “ServicesRegistrySi”.

When you select the “ServicesRegistrySi” proxy, switch to the tab “Configuration” and then to the tab “Logical Ports”. Select the entry “ServicesRegistrySiPort” from the list of logical ports and check whether the “WS Endpoint URL” in the “Details” tab is correct, as shown below:

You can adjust this value by pressing the “Edit” button.
Configure Services Registry proxy (NW CE)

As mentioned at the beginning of this section, the SAP NetWeaver CE 7.1 also comes with an own SAP Services Registry and Enterprise Services Repository. Since it is recommended to use the SAP Services Registry of the SAP NetWeaver PI 7.1, in case you have both systems in your SOA landscape, the SAP NetWeaver CE needs to be configured accordingly.

The configuration steps are the same as in the previous paragraph i.e. configure and test the two destinations “UDDI_DESTINATION” and “CLASSIFICATION_DESTINATION” on the SAP NetWeaver CE system to point to the SAP NetWeaver PI system. It is important to adjust the hostname, port, system name, URL, user and password, as shown below (for the example of the “UDDI_DESTINATION” destination):
Finally, the proxy for the SAP Services Registry should be checked. This procedure is already described in the previous paragraph.

**Configure Visual Composer (optional)**

As a last step, it is necessary to configure the service consumer to be able to access the SAP Services Registry, the Enterprise Services Repository and the back-end system. As discussed earlier, there are several different technologies, platforms and frameworks to consume Enterprise Services, which differ significantly in their set-up procedure. Since this document focuses on SOA Middleware and not on service consumption, we use the Visual Composer as an example service consumption technology.

To set-up the Visual Composer to consume the services listed in the registry, it is necessary to define one more destination of type “Service Registry”, which has the same name as the service providing back-end system, e.g. “CE1”. The following tutorial and this blog describe this configuration step in detail.

**Set-up of non-SAP SOA Middleware Components**

For the demo scenario we installed an additional Enterprise Service Bus, the IBM ESB v6.1. The installation of the IBM ESB can be performed by starting the setup.exe program on Microsoft Windows operating systems. The configuration of the IBM ESB that was necessary to run our demo scenario will be described in the corresponding context later in this blog series.

Another non-SAP component that we used in our example is OpenSSL. OpenSSL is an “Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library” (www.openssl.org). We used OpenSSL to generate all certificates that are needed for the secured communication between the SAP NetWeaver PI and the IBM ESB. Of course, every ESB can generate certificates, but more complex heterogeneous SOA landscapes will certainly have one central certification authority that generates and signs all certificates. We simulate this by using the OpenSSL toolkit. The installation depends on the distribution of OpenSSL. There is a comfortable installer available for Microsoft Windows operating systems. We will describe how to configure OpenSSL and generate certificates in a later blog about using the WS Security standard to encrypt and sign the messages that are exchanged between the two ESBs.