

How to Create a Product for a Composite Application – Part I



Applies to:

NetWeaver Composition Environment 7.20 and higher.

Summary

This paper discusses the concept of a Product for Composite Application development by explaining it from an end to end perspective as well as outlining the advantages of using a Product. It is the first part of a series which has the goal to enable developers and administrators of NetWeaver CE to make use of the Product concept during the development of a Composite Application.

Author: Stefan Henke

Company: SAP

Created on: October 2010

Author Bio

Stefan Henke is a Software Developer/Architect at SAP AG. He has already worked on several projects in the NetWeaver area. Currently, he is working in a feature team of NetWeaver BPM which is part of NetWeaver CE.

Table of Contents

What is a Product?	3
How is SLD involved?.....	5
Why Use the Product?	7
Related Content.....	9
Copyright.....	10

What is a Product?

If you have ever used the Composite Designer perspective in the NetWeaver Developer Studio, you might have come across the term “Product”. You might also have read the blog series by Ivan Stoyanov which describes the set of features of the Composite Designer [1] or my previous article on SDN about Composite Designer [2]. In any of these cases you could be interested in getting some more insights into the Product concept which is provided by the Composite Designer.

But let’s start slowly and take a closer look at what a Product is all about. If you are familiar with java development in the Developer Studio, you will know what a Development Configuration (be it part of a DTR track or just a local one) is about. Such a Development Configuration allows you to include a certain set of Software Components. On the one hand it contains all SCs which you want to develop Development Components in. On the other hand you have to include all SCs of the technologies you want to use in your development. These can be either SCs provided by SAP needed for developing domain specific artifacts (e.g. processes in the Process Composer) or any other 3rd party library. To put it short, the track contains all development artifacts you need to successfully build your Composite Application (see Figure 1 for reference).

SCs are the lowest level of entities that can be shipped and patched on its own. But this is only the technical point of view. Normally, a Composite Application which is shipped and installed consists of several SCs. This brings a lot of benefits when structuring your application and allows you to ship new versions or bug fixes on a smaller granularity than the whole application. If you are using several SCs in your application, the Product acts as an umbrella to bundle these SCs.

The Composite Designer supports the usage of Products in both local and remote Development Configurations. The local scenario does not involve any track in NetWeaver Development Infrastructure (NWDI). So you are able to perform all necessary operations in your local eclipse workspace from within the Developer Studio. This of course makes the Product much easier to use, but you will miss all features for advanced development offered by NWDI. However, to create a Product in the local scenario you can either use the default Development Configuration “Local Development” or your own local Development Configuration created in the Development Infrastructure Perspective. The latter option is interesting because it allows you to use the Product even if you store your code in a 3rd party source management system [3]. Dealing with a Product in a remote scenario is a bit more complex as it requires the usage of CBS and SLD. This series will focus mostly on the remote scenario as firstly it requires much more knowledge about the infrastructure and secondly is the recommended way for development of Composite Applications.

But to avoid misunderstandings: Product is not a term introduced and exclusively used by Composite Designer. Rather it is a general term used in the application lifecycle of SAP software. This is also where the notion of Product in Composite Designer is derived from. One can say the support of a Product in SLD has been extended to the Developer Studio to use it also during the development of an application. You might see this clearly if you are using the Product within a remote Development Configuration.

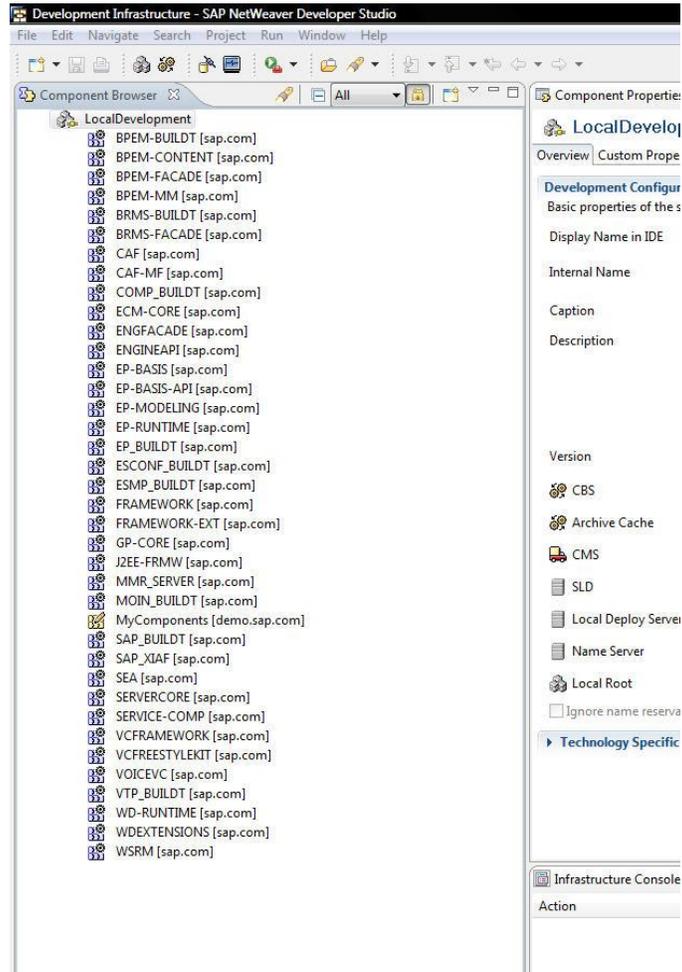


Figure 1: Local Development Configuration

How is SLD involved?

I promised you to go into more detail to explain what the Product in SLD is about and how it is related to the Product in Composite Designer. First of all, let me give you a short introduction into SLD. It acts as the central information storage about a system landscape and allows you to describe the system landscape of your company as well as the installed software on these systems. It is easy to understand that the content is divided into two areas: a Software Catalog and a Landscape Catalog (see Figure 2). The Landscape Catalog contains all technical (physical) and business (logical) systems of your landscape. So, the content of the system catalog is highly individual for every company. The difference between technical and business systems is important especially if you run one physical system with two clients (in German this is the good old “Mandant”). This is treated as one technical and two business systems.

The Software Catalog contains all the software developed by SAP, any partner or even by you. This software is described using Products and Software Components. As a consequence, the Software Catalog contains metadata of all software which is delivered by SAP. But even for software which you are developing on your own (for example using NetWeaver Composition Environment), it is very beneficial to provide the SLD Software Catalog data. This is especially true if you are going to sell this software to other companies. Having registered your software in SLD, it is much easier to support your software when your customer needs any kind of support as you can easily get an overview about what the customer really has installed.

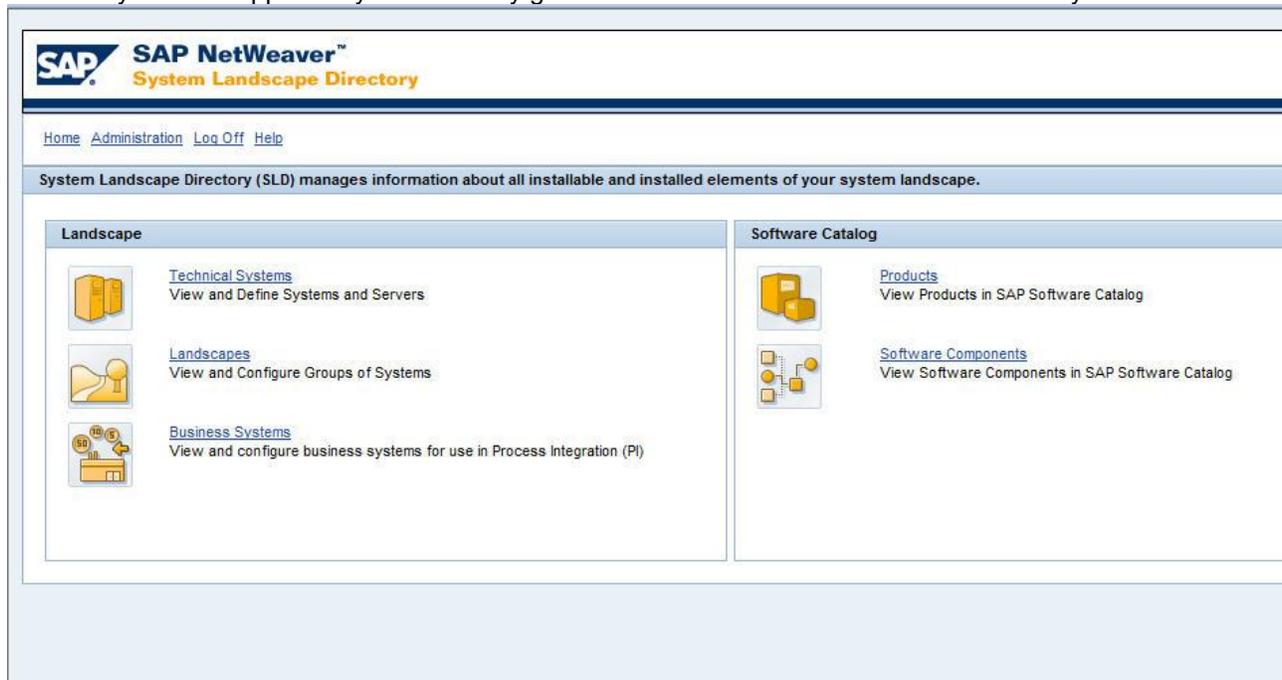


Figure 2: Initial screen of SLD

On a high level, the Software Catalog consists of Products and Software Components. A Product can be seen as an application or set of software that can be bought separately. Such a Product normally consists of a set of Software Components. Figure 3 is showing a screenshot of the Software Components part of the Product “NetWeaver CE 7.2”.

The screenshot shows the SAP NetWeaver CE 7.2 interface with the 'Software Components' tab selected. A table lists the software components for a product instance. The table has three columns: 'Product Instance', 'Name', and 'Software Component Version'. The first row is highlighted in orange.

Product Instance	Name	Software Component Version
SAP NETWEAVER CE 7.2: BPM	16	BPEM ADOBE FLEX UIS 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM BASE 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM BUILDT 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM CONTENT 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM CORE 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM FACADE 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM HUMAN INTERFACE MGMT 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM METAMODEL BASIC LIB 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM NWA PLUG INS 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM PORTAL CONTENT 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPEM WEBDYNPRO UIS 7.20
SAP NETWEAVER CE 7.2: BPM	16	BPM CU UI 7.20
SAP NETWEAVER CE 7.2: BPM	16	BRMS BASE 7.20
SAP NETWEAVER CE 7.2: BPM	16	BRMS BUILD TOOL 7.20
SAP NETWEAVER CE 7.2: BPM	16	BRMS CORE 7.20
SAP NETWEAVER CE 7.2: BPM	16	BRMS FACADE 7.20
SAP NETWEAVER CE 7.2: BPM	16	BRMS WEBDYNPRO UIS 7.20
SAP NETWEAVER CE 7.2: BPM	16	BUSINESS RULES MONITORING 7.20
SAP NETWEAVER CE 7.2: BPM	16	CAF MF 7.20
SAP NETWEAVER CE 7.2: BPM	16	CAF UI PATTERNS 7.20

Figure 3: List of Software Components for Product

Even if the Landscape and Software Catalog are somehow separated, it is important to understand that SLD allows you to link content from both catalogs. This makes it possible to map a certain piece of software to a system in your landscape. This does not only give you an overview of the landscape in one central place, but also simplifies integration of applications and systems as the SLD can be queried by many integration components (e.g. Process Integration, Solution Manager, NWDI). The NWDI (in particular CBS and CMS) uses SLD data to set up a track for team development, but also leverages the data to organize transports of Software Components between different systems (e.g. from dev system to test or productive system). We will use this capability later on.

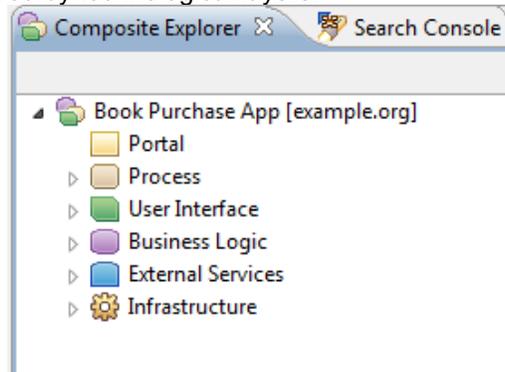
In a development environment that is used for developing productive software, the usage of NWDI is recommended by SAP. In the past, even if you had already defined your application as Product and Software Components in SLD, the structure of the Product got lost when coming to NWDI and development in NWDS. This has now been changed by extending the support of Products to the NWDS (in Composite Designer). In the end it boils down to improving the application development by bringing the Product metadata from SLD into the NWDS.

In the next section I will give some more details on how these improvements look like and how you might leverage them in your development.

Why Use the Product?

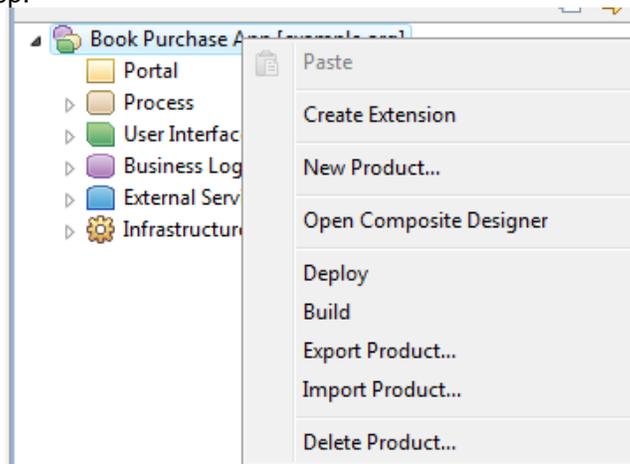
The previous section has already given some overview and motivation why the usage of a Product is beneficial. This part will describe these advantages in a more technical depth.

As mentioned above, a Product acts as an umbrella for development of Composite Applications in the NWDS. It allows you to work on one Composite Application as one unit even if the components are spread across multiple Software Components. This umbrella is visible in the Composite Explorer which is part of the Composite Designer. Each Product is displayed as a root element providing all development components and artifacts in its sub tree grouped by technological layers.



The Composite Designer provides an editor (so called Composite Overview) which can be opened for a Product. Here, you can nicely see all development artifacts of your Composite Application in a graphical manner. In order to more easily get the big picture, the objects are arranged according to the technology layer they belong to. In both cases, the DCs and/or development objects are strictly grouped by technical layers starting from portal on the top to business logic and external services resp. infrastructural parts in the bottom.

Using a Product during the development phase allows you to trigger lifecycle operations for the whole Composite Application rather than for single Development Components or Software Components only. This includes creation, deletion, build and deployment. Rather than doing this for each and every Development or Software Component of your application, the Composite Designer provides support for these things on a Product level in a single step.



If you have ever had to build and deploy a big application to the java server, you will know what I mean. It can be very painful to do this for each and every DC. But even more interesting is the creation of a Product. Using the “New Product” wizard provided by Composite Designer, it is possible to create a Composite Application and immediately define its overall structure in terms of Software Components and most important the technologies you want to use. Defining the set of required technologies will automatically create the necessary Development Components in your Product.

Just for completeness, as there are some (important) differences between the Product related functionality in local and remote development Table 1 is outlining these differences.

	Local Development	Remote Development (Track based)
Visibility of Product in Composite Explorer	X	X
Composite Designer Overview Editor for Product	X	X
Product with multiple SCs	X	X
Import / Export Product from/to filesystem	X	X
Product creation	X	X (with limited set of features; see below)
Automated DC creation for required technologies in New Product Wizard	X	X
Define SC structure in New Product Wizard	X	Only via CBS and SLD by Administrators
Delete Product	X	X (only deletes it locally from the workspace)
Deploy Product	X	X

Table 1: Features of Product Concept

In the next part of this article series, I will give a step by step tutorial how to create and use a Product in Composite Designer. So, stay tuned.....

Related Content

<http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/17417>

<http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/f04a0a13-0ca1-2c10-0dbc-b5e131982f62>

http://help.sap.com/saphelp_nwce72/helpdata/en/49/11e88ccf292810e10000000a42189c/content.htm

Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.