The Fast Way to Component-Based Development Using SAP NetWeaver CE 7.1 – Part I: Concepts & Development Scenarios

Applies to:

SAP NetWeaver Composition Environment 7.1.

Summary

With its most recent release of SAP NetWeaver Composition Environment 7.1 (CE), SAP is now a certified vendor of a JEE 5.0 application server. In addition to delivering a pure implementation of the standard, SAP has added several enhancements of concepts and object types to the Java development. These can optionally be used to make development more efficient, or to better integrate with the SAP ecosystem. When it comes to object types, the enhancements address certain development issues, such as a client-independent UI, the handling of database tables, or the creation of composite applications. Whilst the concepts enhancements ensure an improvement in the structuring of applications and the development process.

As a customer or partner who wants to deliver objects targeting SAP systems, you are free to use either SAP's concepts and objects, or to continue with the pure standard J2EE/JEE when developing for the SAP NetWeaver platform. Using the SAP NetWeaver Development Infrastructure (NWDI), SAP has been providing a comprehensive approach to the complete development life cycle of Java-based applications to support development for years.

With SAP NetWeaver CE, SAP is starting a new development scenario that, with minimum cost and effort, allows you to integrate a pre-existing infrastructure with SAP's development processes and ecosystem You just have to choose which scenario you want to use.

Author: Wolf Hengevoss

Company: SAP

Created on: 21 November 2007

Author Bio

Wolf Hengevoss graduated in natural sciences from the University of Kaiserslautern. In 1999 he joined SAP as a member of product management of the Basis group working on topics such as Computer-Aided Test Tool and Business Address Services.
Since the early stages of SAP Exchange Infrastructure, he has been working on the Java environment. Today, his focus is on SAP's Software Change Management of non-ABAP applications.

Table of Contents

Introduction

SAP provides comprehensive tools for Software Change Management processes for both ABAP and Java-based development. You will find a central development infrastructure for both application servers: For ABAP developers, all parts, such as the development environment, server-side development processes, and runtime, run on the same SAP NetWeaver system. Whilst for Java, the SAP NetWeaver Development Infrastructure (NWDI for short) is a separate server installation that is called by many IDEs (installed locally) and which centrally manages many software projects on separate servers.

Despite these fundamental differences, the two approaches do share a common task – that of supporting developers and administrators during the whole development life cycle. The most prominent of all the features of the NWDI is the strive for a consistent system state management for development systems and runtime that is ensured by centrally managing source files *and* build artifacts.
For Java development, a set of metadata can be added to Eclipse-projects that to define the inner structure of applications – that is what we call SAP's component model. One practical outcome of using the component model is that (many) SAP Java applications allow modifications to be undertaken at customer sites and can also be managed by update processes - a key goal for the development of the NWDI. If you are a software vendor who is independent from SAP, this open approach to development allows you to integrate into the SAP ecosystem.

So SAP offers a development infrastructure – however, you may already have an infrastructure in place that you want to use – you must decide if you will continue with it, or if you will move to the NWDI. Another question is: Do you want to continue with pure J2EE/JEE development for the SAP NetWeaver platform, or do you want to use some of SAP's object types also? You have many options with SAP NetWeaver CE: Deliver pure standard applications, dive into component-based development using your infrastructure, or invest in SAP's comprehensive development infrastructure.

This is where the story begins…

Development Scenarios with SAP NetWeaver CE 7.1

Java-Based Development at SAP – a Glossary

Let us start with a few facts you should know about Java-based development at SAP – you should be informed about what is on offer before you make your choice:

- Apart from SUN, of course, SAP was the first certified vendor of a ***JEE 5.0*** server implementation so the environment for Java-based development is right up-to-date.
- SAP's IDE (SAP NetWeaver Developer Studio) is based on the **open Eclipse framework**. SAP's IDE connects to the central development infrastructure using HTTP(S)-based communication, WebDAV and DeltaV implementation for source code versioning, and also using an ANT-based build process – all open standards again.
- So why, you may ask yourself, did **SAP create an IDE and development infrastructure** of its own? The answer to this is because some key issues have not been addressed by those tools available on the market, these include a robust build process, a way to organize the reuse of objects at a project level, and modification management – or concurrent development in general.
- Additionally, some object types, for using databases in Java development or composite applications for example, were missing. For these new objects one more idea comes into play: SAP's component model, adding metadata to projects, is breaking applications down into reusable components that contain ***development objects***, such as class files and so on. These components work as black boxes that interact via defined interfaces only. As one of the benefits of this encapsulation of functionality using the component model allows for a build process on the granularity of these ***development components*** (with the granularity of Eclipse projects) instead of a build of a complete application. Development components are grouped into ***software components*** for installation and upgrade, these are aggregated into ***products***, which you the sell or buy. The component model's key element is that it clearly defines dependencies between components.
- While all the standard Java and JEE projects can still optionally be used without **SAP's component model** in SAP NetWeaver, it is used for many of the SAP-specific types: The component model is a prerequisite for Composite Applications and at least highly recommended for Web Dynpro development – for the latter you may create this project type without the component model as long as you do not want to use more than one Web Dynpro project in your application.
- Intended to be used by big development teams even if their members are distributed across different locations, a management concept for development was needed to ensure success: We call this ***development configurations***, a description of a development environment written as an XML file, which is then used to control the IDE's access to the relevant sources and artifacts. Development configurations are used to organize component-based development. As you will see later, they can now be created either centrally in the NWDI or locally. Everyone working with the same development configuration starts with the same set of objects, which has great advantages for syncing development efforts. The result of working in a development configuration is that a software component archive (SCA) is formed, no matter if working with, or without, central services.
- The services of the **NWDI** provide a server side development landscape for component-based development with source code control in the Design Time Repository (DTR), artifact and build management (Component Build Services or CBS), and administrative control in the Change Management Service (CMS). Development according to JEE 5.0 is fully supported in the NWDI.

Note: These are just a few ideas you should be aware of in preparation for the rest of this article.
If you want to learn about the NWDI-based development in detail, refer to the NWDI pages in SDN to learn about the processes, tools, and tips & tricks under SAP NetWeaver Capabilities → Lifecycle Management → Software Logistics → Software Change Management → NWDI

Java-Based Development Scenarios at SAP

Firstly we need to discuss what *you* need from all of these concepts. It might very well be the case that you are already running a non-SAP development process. In any case, you will still need to have versioning of sources, take care of the build process, and manage development systems, but you might want to continue with the tools that you are used to – is that possible? The answer is "Yes, but…" – in the end it depends on how deeply you want to dive into SAP's approach to Java development. To cut a long story short let me give you an overview of the possible scenarios and their most important features in the table below.

As you can see, the list of technologies you can use grows when you move from the "pure standards" scenario into component-based development – at first you start with the standard types and then you get all the others in the following step. The distribution model is totally your and your customers' responsibility. You can use SAP Service Marketplace to obtain certain pieces of information for your, and your customers', systems, including software catalog updates from the SLD and updates of the SAP software components that you use.

Table 1 compares the requirements and benefits of the three scenarios:

Scenario **Features**	**Standard Development**	**Component-Based Development with Optional Infrastructure**	**Component-Based Development with Full-Blown SAP NetWeaver DI**
Pre-requisites	SAP NetWeaver CE Developer Studio + RTS Optional use of Dev. Infrastructure (SAP or 3rd Party)	SAP NetWeaver CE Developer Studio + RTS + (local) development configuration Optional use of Dev. Infrastructure (SAP or 3rd Party)	SAP NetWeaver CE Developer Studio + RTS SAP NetWeaver DI – release 7.0, SPS13 is highly recommended
Object Types	All standard J2EE/JEE types	All standard J2EE/JEE types All SAP-specific types	All standard J2EE/JEE types All SAP-specific types
Key Benefits	Very low investments in new processes (learning, systems)	Low investments Few new processes required, easy learning Local Development Configuration handling	Completely integrated development process Modification concept in place
Keep in Mind	None of the SAP-specific object types can be used No integration into SAP's modification concept	No complete integration into SAP's modification concept	Higher investments in new processes required (learning, systems)

Table 1: SAP NetWeaver CE – Comparison of Development Scenarios

The following will concentrate on the scenario "Component-Based Development with Optional Infrastructure": It is hard to write about the "Standard-Development" scenario as it does not make many assumptions about how you organize your work, whilst development with SAP NetWeaver CE 7.1 is fully supported by the NWDI of SAP NetWeaver 7.0 SPS 13 using the same comprehensive process that has been established at SAP, customers, and partners for many years. You will find it described in SDN and the SAP Help Portal, so there is nothing new to tell – however, if you are unfamiliar with this then I recommend that you follow the link to the NWDI pages in SDN at the end of this article.

Component-Based Development with Optional Infrastructure

You have just read that standard JEE development is part of the SAP NetWeaver CE. The simple approach is to settle for JEE development only. However, there are some non-standard features that are worth noting:

- There are Eclipse project types that allow you to create database tables (type *Dictionary*), provide an interesting UI framework (type *Web Dynpro*), or a composite application (type *CAF*), and so on. These are non-standard types. These objects have in common that they are wholly or partly based on SAP's component model mentioned before.
- Another aspect – also related to the component model – is that you run into problems if local development environments in the development team differ from each other: If you put an object into the compilation process using an outdated version of another object also under development elsewhere, this will cause trouble. To avoid this, the starting point of development should be the same for everybody in the development team pursuing the same goal.

For hundreds of SAP developers this means that they rely on the NWDI for the central management of sources, artifacts, and deployment, organizing the development landscape with development configurations hosted by the server infrastructure. However, if you are a Software Service Provider (SSP) independent from SAP, you might be looking for a more lightweight solution or you might already have one that you do not want to change. Our goal was to enable you to use – in a low-consumption-compatible way – your existing development infrastructure tools and processes whilst building applications with and for the SAP NetWeaver Composition Environment.

Development Process Overview

This gives you the big picture of what to expect in this scenario – the detailed steps will be described in the next part of the article.

Defining the Development Environment – Development Configurations

If you are going to provide software to SAP's customers, they might expect objects that are tailored to meet the requirements of their SAP NetWeaver runtime systems and make use of those SAP project types mentioned above. As we have seen, many of these need the component model and development configurations. Therefore, the question for you might be: How do I get into SAP's component model in the easiest way with the smallest investment? The answer to that is to create development configurations locally yourself without the central systems. This enables you to use – in a low-consumption and compatible way – the SAP NetWeaver Developer Studio in combination with your existing development infrastructure tools and processes whilst building applications with SAP NetWeaver Composition Environment. I will describe this process below.

You can use a development configuration on a single PC to create component-based applications for SAP NetWeaver CE. More probably, however, you will work in a team. All you need to do is distribute the development configuration to your team and to use the file system to exchange data.

Sharing Development Objects in the File System

The file system is the minimum common denominator in your team. Working together requires at least a file share for team members. The next natural step is to put this under version control using any tool you like, for example CVS, or see SAP's approach to file versioning the Design Time Repository (with versioning mechanisms that allow modification scenarios), which can be used independently of the NWDI.

Setting Up a Central Build Process

Anything you produce in the Developer Studio, you can build there. This includes, of course, development components. However, if you are working in a team producing components that depend on each other, you will need to have a complete selection of all objects – source files and artifacts – available for building the complete application. For this purpose, SAP delivers a command line tool to deal with the build centrally in the file system the NWCETOOL.

Shipping the Software – Creating Software Components

Once all components of an application have been created and tested, you need a format to deliver and install the application: The format we are looking for is exactly the one we started with when we created a development configuration: We produce one or more SC archives or SCA-files. These are the units of delivery, installation, and upgrade. The tool to produce these units is again the NWCETOOL.

Organizing the Next Steps – Maintenance and Enhancement of Applications

SCA files, let me add, can also be reused in the next development configuration. You can either use SCA files to create the next version of the software whilst continuing with development, or set up a development configuration for maintenance purposes, or you can even use the new SC as a basic layer for other functions created on top of it – in this case it will become a required SC in the new configuration.

Conclusion

If you want to deliver JEE applications targeting SAP NetWeaver CE runtime systems, but do not need a complete infrastructure for your work – or you are already using one – the scenario described here is what you are looking for. It provides you with a set of tools to create, build, and package applications according to SAP's component model. All these functions are delivered with the SAP NetWeaver Developer Studio so system requirements and installation efforts are minimal. If you are interested in further details, read part II of this article describing the steps of the development process in detail - expected mid December.

Related Content

You will find related SDN documents or web pages under the following links:

- SAP NetWeaver Development Infrastructure, which brings in-depth discussions of the NWDI Services and the component model
- The enhanced Change and Transport System is capable of non-ABAP transports – very likely it will be in use at your customers' site and might be the link between your delivery and their productive systems.

Copyright