

NWDI Empowered Landscape vs. Landscape without NWDI Setup



Applies to:

Application Lifecycle Management, Composition Environment (CE), Enterprise Portal (EP), Product Lifecycle Management, SAP NetWeaver Platform. For more information, visit the [Application Management homepage](#).

Summary

This article explains about the differences between NetWeaver Development Infrastructure (NWDI) empowered landscape and Landscape without NWDI Setup. Also, why NWDI empowered landscape is more superior and its advantages over Non-NWDI Landscape.

Author: Shreyas Pandya

Company: Enteg Infotech Pvt. Ltd.

Created on: 3 February 2011

Author Bio



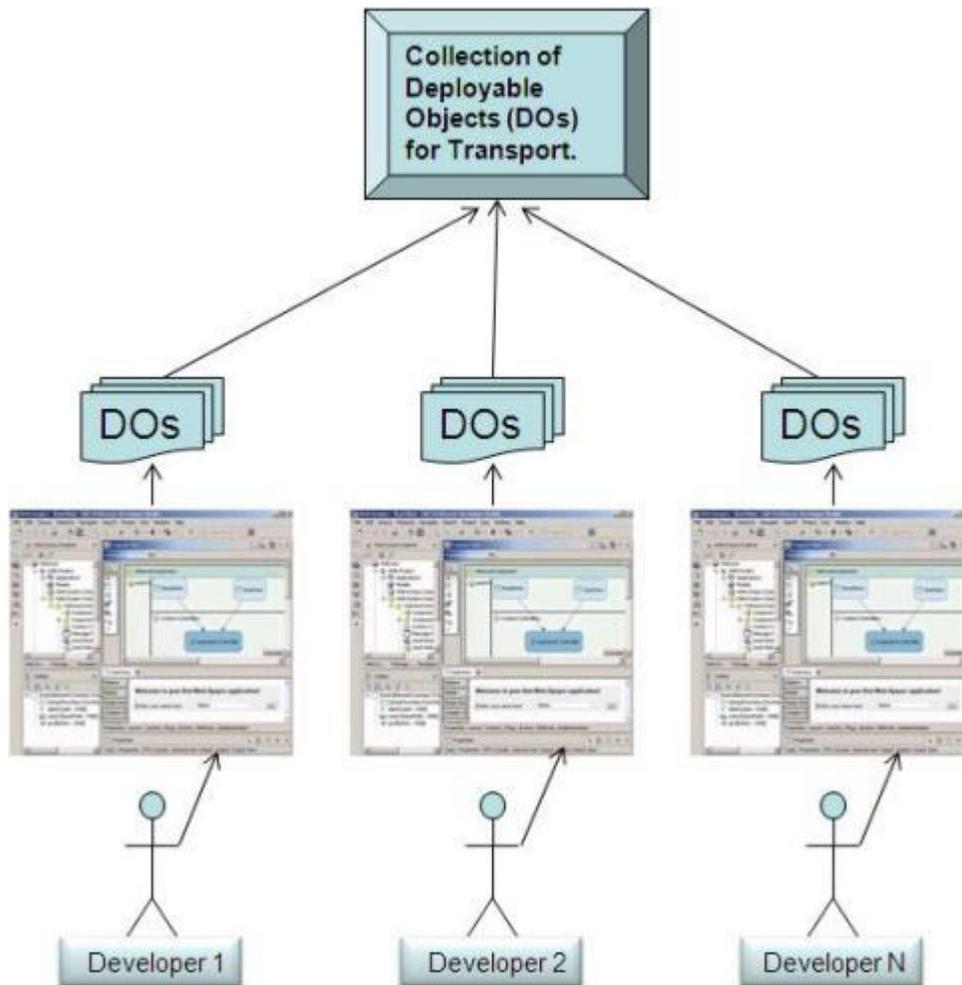
Shreyas Pandya is a SAP Netweaver Enterprise Portal Consultant at Enteg Infotech Pvt. Ltd, India. His areas of expertise in SAP technologies include Netweaver Development Infrastructure (NWDI), WebDynpro Java, Composition Environment, and Enterprise Portal

Table of Contents

Landscape without NWDI Setup	3
Disadvantages of Landscape without NWDI Setup	3
Building Blocks of SAP's Component Model:	4
NWDI Empowered Landscape:	5
Advantages of NWDI Empowered Landscape:	5
Related Content	6
Disclaimer and Liability Notice	7

Landscape without NWDI Setup

1. Individual Developers work on their local development environment in Developer Studio.
2. In case of big projects, each developer in a team works on different DCs (Unit of Reuse) assigned to them separately.
3. Output of development is deployable objects (like .EAR files), which are collected manually at the end of the development phase for transport.

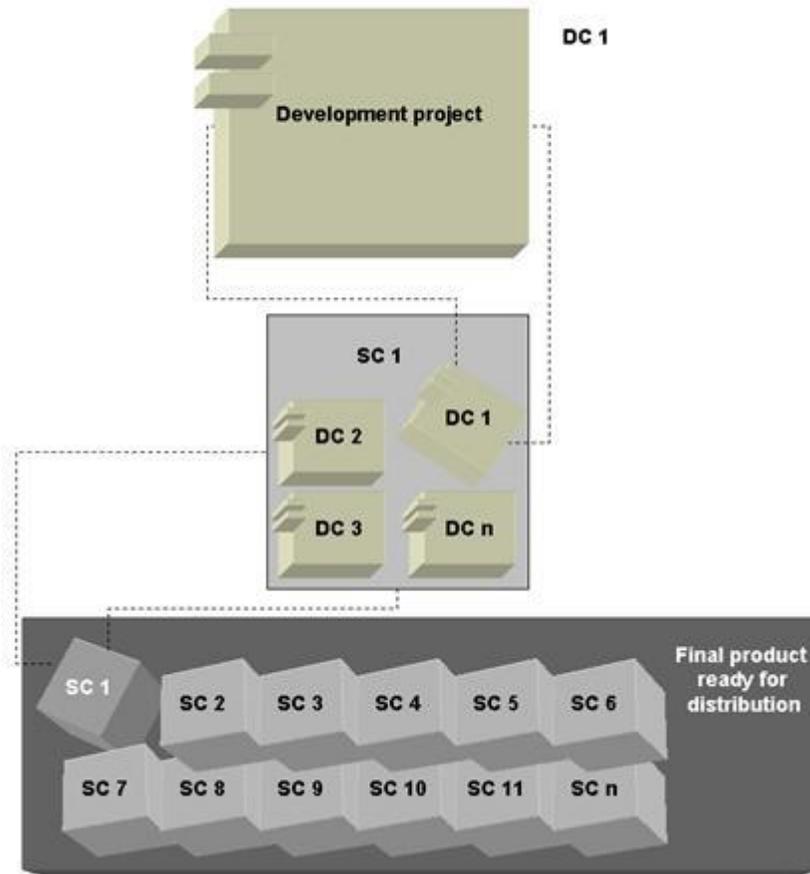


Disadvantages of Landscape without NWDI Setup

1. Entire development takes place in a scattered pattern, i.e. all the development objects are dispersed amongst the development team members.
2. There is no centralized management of the source code.
3. Multiple developers cannot work on a single development object simultaneously due to lack of version control.
4. There is a possibility of missing some of the development objects during the transport operation, as it has to be collected manually from all the developers (Human Intervention is involved), which could possibly result into a dirty transport.
5. This kind of development model doesn't fully comply with the SAP's Component Model, which recommends Development Component (DC) usage as a standard practice.

Building Blocks of SAP's Component Model:

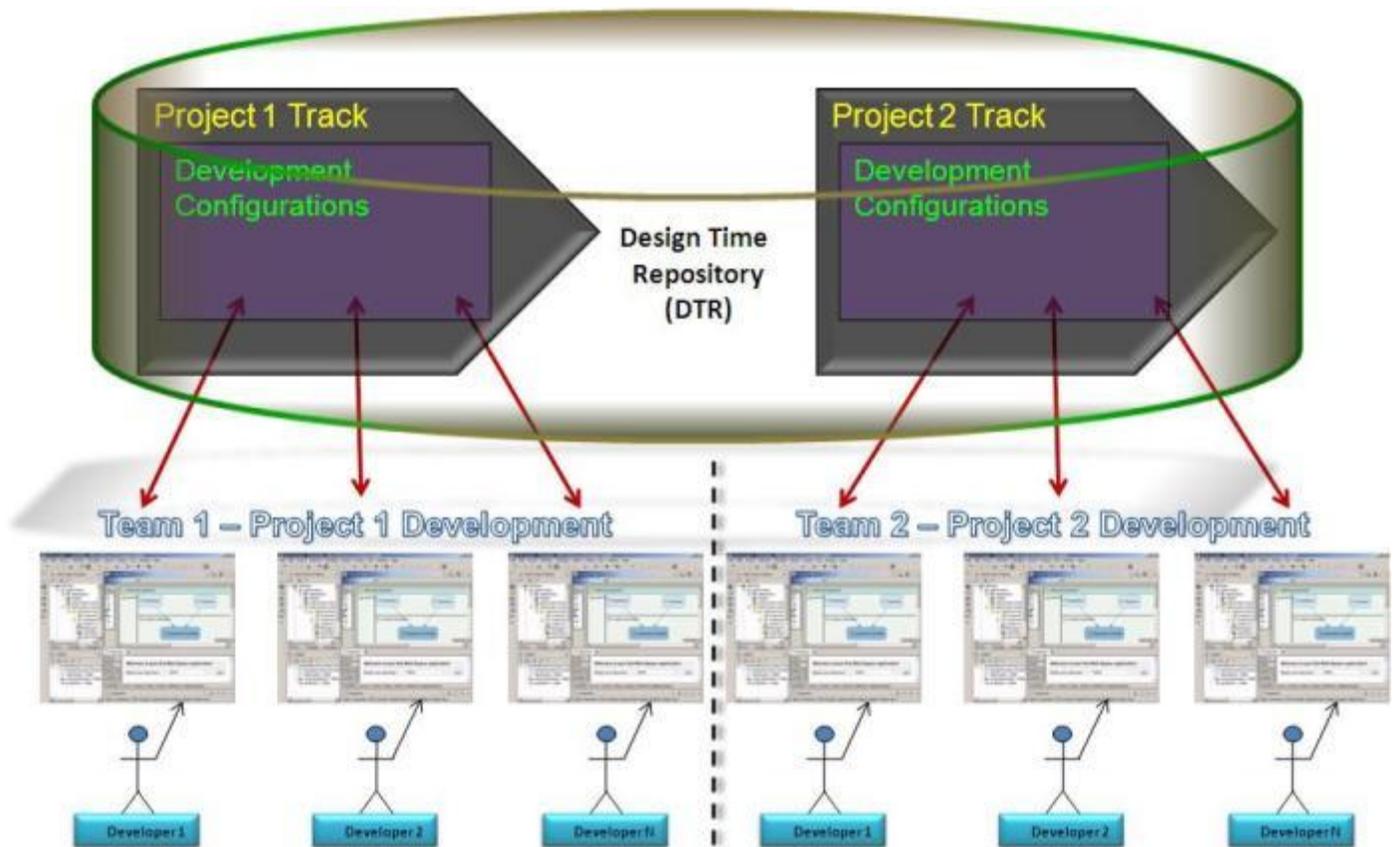
By dividing software projects into components, you can organize them right from the start into comprehensible and reusable units. Components can use other components in a well-defined and controlled manner, can encapsulate child components, and can publish their functions in a set of public interfaces called public parts. These features are applied in the component model.



1. **Product - Unit of Delivery:** A piece of software that you run or sell to a customer. Products consist of software components (SCs).
2. **Software Component (SC) – Unit of Installation and Transport:** Unit of Transport and Installation: An installable piece of software. SCs contain development components (DC) and are (re-)used in products. SCs "know" their dependencies to other SCs.
3. **Development Component (DC) – Unit of Reuse:**—A piece of software on the granularity of an Eclipse-project, containing all the development objects like Java classes, interfaces, etc.
4. **Development Objects:**—Smallest unit in component hierarchy, example Java Classes, Interfaces etc...

NWDI Empowered Landscape:

Revolutionized Development Paradigm



Advantages of NWDI Empowered Landscape:

1. Entire development takes place in an organized pattern in a symphony, i.e. all the source code is managed in a centralized repository called Design Time Repository (DTR).
2. Multiple developers can work on a single development object simultaneously, due to presence of version control.
3. Unlike classical way of transport described in Landscape without NWDI Setup, there is no possibility of missing any of the Deployable Objects during the transport process because, it is an automated process that is taken care by CMS (Change Management Service) in which .SCA file is constructed as a single deployable unit.
4. Each Project has a well defined boundary because, before the development is initiated, the Landscape Configurator prepares a "Track" for the Project and delivers it to the developer(s) so that developer(s) can only focus on the development configurations belonging to their bucket.
5. This kind of development model fully complies with the SAP's component Model. Which best fits the easy maintenance and re-usability concepts.

Related Content

[SAP NETWEAVER DEVELOPMENT INFRASTRUCTURE](#)

[NWDI-based Development Process](#)

[Best Practices for NWDI: Track design for ongoing development](#)

For more information, visit the [Application Management homepage](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.