

**How-to Guide
SAP NetWeaver '04**



**How to ...
Get started
with the
SAP NetWeaver
Development
Infrastructure**

Version 1.00 – July 2006

**Applicable Releases:
SAP NetWeaver '04**

© Copyright 2006 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation 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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines /strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

1	Introduction.....	2
1.1	Prerequisites	2
1.2	Overview.....	2
2	Configuring a Product and Tracks.....	3
2.1	How to Create a Product.....	3
2.2	How to Create a Software Component.....	5
2.3	How to Reserve the Namespace Prefix	8
2.4	How to Create a Domain.....	10
2.5	How to Create a Track.....	11
2.6	How to Create Track Connections (Optional).....	16
2.7	How to Check In and Import Required Software Components.....	20
3	Development – Working with the Component Model.....	22
3.1	How to Import the Development Configuration from SLD	22
3.2	Developing in SAP NetWeaver Developer Studio	25
3.3	How to Create Public Parts of a DC.....	29
3.4	How to Use a Public Part (Defining Usage Dependency).....	31
3.5	How to Check In and Activate	32
3.6	Conflict Resolution	34
3.7	How to Release Components.....	36
4	Transporting Components.....	37
4.1	How to Import to the Consolidation System.....	37
4.2	How to Assemble.....	38
4.3	How to Import the Archives to the Test System.....	39
4.4	How to Approve (Approval process).....	40
4.5	How to Import to the Production System	41
5	Appendix A: Steps for Creating a and Assigning it to a User.....	42
5.1	Create a Role.....	42
5.2	Create a Group.....	43
5.3	Assign a Group to a Role.....	43
5.4	Assign a Group to a User	45
6	Appendix B: Template Installer.....	47

1 Introduction

This document covers the process cycle of Administration and Development using the Development Infrastructure. The steps described in this document are based on Release SAP NetWeaver 04 SP16.

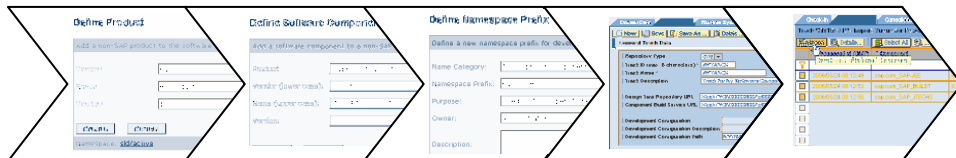
1.1 Prerequisites

- Complete installation of SAP NetWeaver Application Server Java
- Complete installation of DI (CMS, DTR, and CBS) on SAP J2EE Engine
 - *Refer to the following link for details of installation and post-installation of the Development Infrastructure:
Go to the Service market place [<http://service.sap.com/instguides>] and choose NetWeaver -> Release<release version> -> Installation -> DevEnv
- System Landscape Directory (SLD) installed as a Name Server

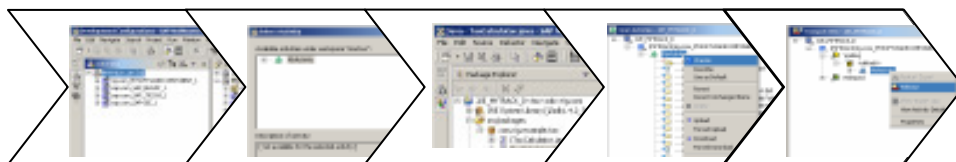
1.2 Overview

The process can be separated into roughly three sections. The initial process is the preparation phase by the administrator. During this phase, the configurations, such as creating *products*, *software components*, *tracks* need to be completed. The next phase is the development phase. This phase is processed by each developer. After the development tasks are completed, the administrator needs to perform the transport and assembly. This final phase enables the software developments available to the production system. This cycle can be reiterated for software maintenance. The following figure roughly describes the flow of the steps covered by this document.

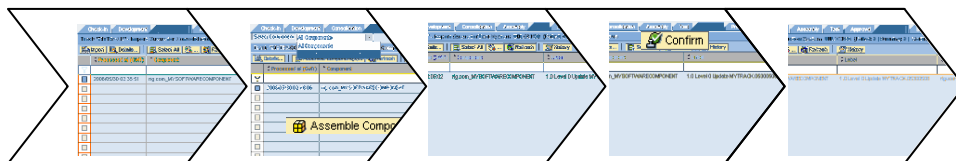
Configuration (Administrator)



Development (Developer)



Transport (Administrator)



2 Configuring a Product and Tracks



2.1 How to Create a Product



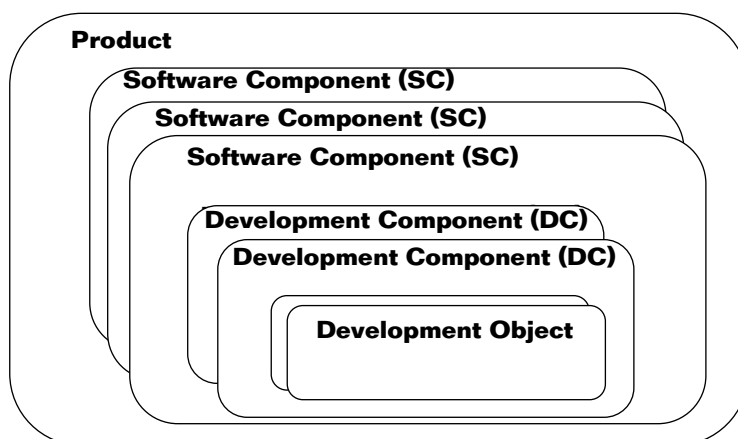
2.1.1 Role

Administrator (Assign the user to the “Administrators” Group)

* The “Administrator” role must be assigned to the “Administrators” Group.

2.1.2 Goal

SAP’s java-based development is based on the structure of the SAP component model. The component model is structured as in the graphic below. Development starts from defining the *product* in the System Landscape Directory (SLD). In this section, you will create a *Product*.

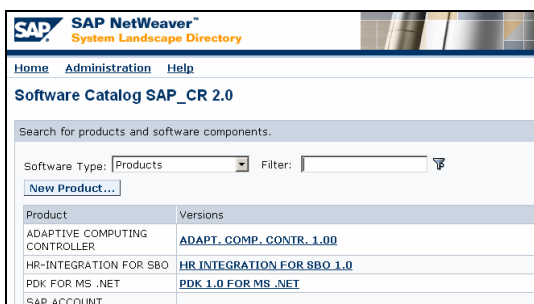


2.1.3 Steps

Open the System Landscape Directory (<http://<hostname>:<port>/sld>) and choose the “Software Catalog” link.



You will see a list of the products and software components in the system landscape. Choose **New Product...** to create your own *Product*.



Add the *Vendor*, *Name* and *Version* and choose **Create** to complete the creation process.

Define Product

Add a non-SAP product to the software catalog.

Vendor:

Name:

Version:

Namespace: sld/active J2E



Products are what you run or sell to a customer. The internet domain (e.g. sap.com) of the owning company is used to indicate the product ownership. You therefore need to enter the internet domain under *Vendor*.

Under *Version*, you have to enter a numeric value.

2.2 How to Create a Software Component



2.2.1 Role

Administrator (Assign the user to the “Administrators” group)

* The “Administrator” role must be assigned to the “Administrators” group.

2.2.2 Goal

Software Components (SCs) is a definition to the *Development Components (DCs)* group. With a software component, you define a container which is meant to contain a specific set of functions to be used in one or several products, as well as the definition of the dependencies to other SC version. In this section, you will create a Software Component.

2.2.3 Steps

After section 2.1, you will be directed to the step for creating the *Software Component*.

Fill in the entries and choose **Create** to finish creating the Software Component.

Choose **Cancel** and return to the software catalog list.

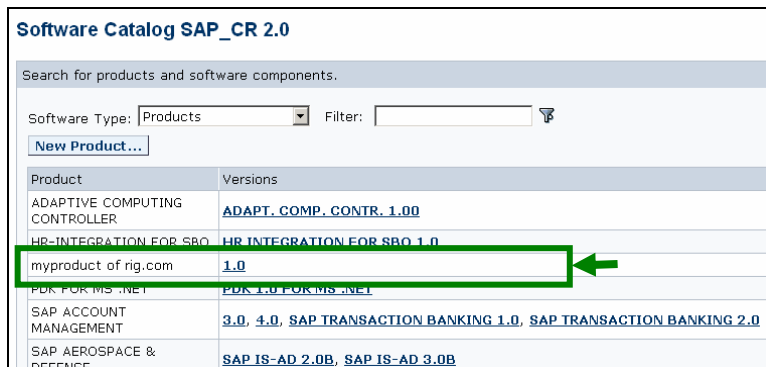
Each SC contains the information about the SCs with which it has a dependency. Here, you need to define the dependencies of your Software Component that are required/used by the central build process. The required software components are as follows:

- | | |
|---|-------------------------------------|
| <input checked="" type="checkbox"/> SAP J2EE ENGINE <Release> | (SAP-JEE <Release>) |
| <input checked="" type="checkbox"/> SAP BUILD TOOL <Release> | (SAP_BUILD <Release>) |
| <input checked="" type="checkbox"/> SAP JAVA TECH SERVICES <Release> | (SAP_JTECHS <Release>) |

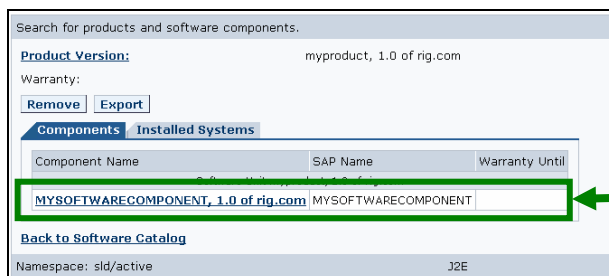


The three SCs listed above are basic sets that you always need to define. Certain applications may require additional dependencies.

Choose the entry in the “Versions” column for the Product that you created in section 2.1.




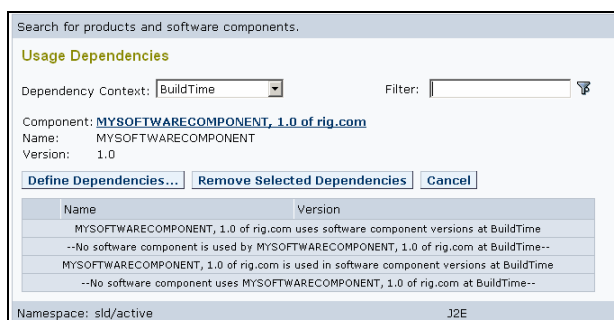
Choose the Software Component name and call up the detailed screens where you can add new dependencies.



Choose the “Usage Dependencies” link.



Choose the “Define Dependencies...” button to display a list of all the software components. Enter “SAP J2E*” in the filter field and choose the  icon.



Select the checkbox for the relevant SAP J2EE Engine version and choose **Create** to define the dependencies.

Search for products and software components.

Define Dependencies

Select for context **BuildTime** the software component versions that are **prerequisites** for component version **MYSOFTWARECOMPONENT, 1.0 of rig.com**.

Create **Cancel** Filter: **SAP J2E***

Filter dependencies from context InstallationTime:

Software Component Version	Name	Version
<input type="checkbox"/> SAP J2EE ENGINE 6.10	SAP-JEE	6.10
<input type="checkbox"/> SAP J2EE ENGINE 6.20	SAP-JEE	6.20
<input checked="" type="checkbox"/> SAP J2EE ENGINE 6.40	SAP-JEE	6.40
<input type="checkbox"/> SAP J2EE ENGINE 7.00	SAP-JEE	7.00
<input type="checkbox"/> SAP J2EE ENGINE CORE 6.40	SAP-JEECOR	6.40
<input type="checkbox"/> SAP J2EE ENGINE CORE 7.00	SAP-JEECOR	7.00
<input type="checkbox"/> SAP J2EE ENGINE SIN. NODE 6.20	SAP-JEESN	6.20
<input type="checkbox"/> Select All 7 Items		

Repeat these steps to add two more dependencies to your SC. Use the following filter values to search the target dependencies:

Filter value: **"*BUILDT*"**

JDI BUILD TOOL <release>

Filter value: **"*TECH*"**

SAP JAVA TECH SERVICES <release>

Search for products and software components.

Define Dependencies

Select for context **BuildTime** the software component versions that are **prerequisites** for component version **MYSOFTWARECOMPONENT, 1.0 of rig.com**.

Create **Cancel** Filter: ***BUILDT***

Filter dependencies from context InstallationTime:

Software Component Version	Name	Version
<input type="checkbox"/> BI BUILD TOOL 3.50	BI-BUILDT	3.50
<input type="checkbox"/> BI BUILD TOOL 7.00	BI-BUILDT	7.00
<input type="checkbox"/> DI BUILD TOOL 7.00	SAP_BUILDT	7.00
<input type="checkbox"/> DI BUILD TOOL 7.10	SAP_BUILDT	7.10
<input checked="" type="checkbox"/> JDI BUILD TOOL 6.40	SAP_BUILDT	6.40
<input type="checkbox"/> MI BUILD TOOLS 7.00	MIBUILDT	7.00
<input type="checkbox"/> Select All 6 Items		

Search for products and software components.

Define Dependencies

Select for context **BuildTime** the software component versions that are **prerequisites** for component version **MYSOFTWARECOMPONENT, 1.0 of rig.com**.

Create **Cancel** Filter: ***TECH***

Filter dependencies from context InstallationTime:

Software Component Version	Name	Version
<input type="checkbox"/> BP CH MGMT CRM HIGH TECH 60.1	BP_CH_MGMT_CRM_HIGH_TECH	60.1
<input type="checkbox"/> MDM TECHNOLOGY 700	MDM_TECH	700
<input type="checkbox"/> MULTI LANGUAGE TECHNOLOGY 6.40	DI_MLT	6.40
<input type="checkbox"/> MULTI LANGUAGE TECHNOLOGY 7.00	DI_MLT	7.00
<input type="checkbox"/> MULTI LANGUAGE TECHNOLOGY 7.10	DI_MLT	7.10
<input checked="" type="checkbox"/> SAP JAVA TECH SERVICES 6.40	SAP_JTECHS	6.40
<input type="checkbox"/> SAP JAVA TECH SERVICES 7.00	SAP_JTECHS	7.00

You have now defined the three mandatory dependencies **SAP_BUILDT**, **SAP-JEE**, and **SAP_JTECHS** to your software component.

Search for products and software components.

Usage Dependencies

Dependency Context: **BuildTime** Filter:

Component: **MYSOFTWARECOMPONENT, 1.0 of rig.com**
 Name: MYSOFTWARECOMPONENT
 Version: 1.0

Define Dependencies... **Remove Selected Dependencies** **Cancel**

Name	Version
MYSOFTWARECOMPONENT, 1.0 of rig.com uses software component versions at BuildTime	
<input type="checkbox"/> JDI BUILD TOOL 6.40	SAP_BUILDT 6.40
<input type="checkbox"/> SAP J2EE ENGINE 6.40	SAP-JEE 6.40
<input type="checkbox"/> SAP JAVA TECH SERVICES 6.40	SAP_JTECHS 6.40
MYSOFTWARECOMPONENT, 1.0 of rig.com is used in software component versions at BuildTime	
--No software component uses MYSOFTWARECOMPONENT, 1.0 of rig.com at BuildTime--	
<input type="checkbox"/> Select All 6 Items	

2.3 How to Reserve the Namespace Prefix



2.3.1 Role

Administrator (Assign the user to the “Administrators” group)

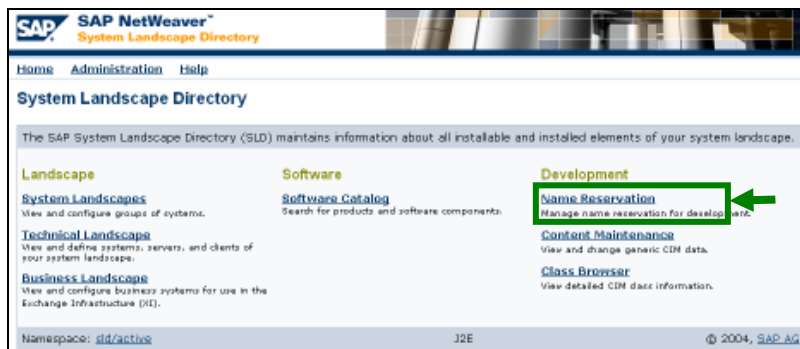
* The “Administrator” role must be assigned to the “Administrators” group.

2.3.2 Goal

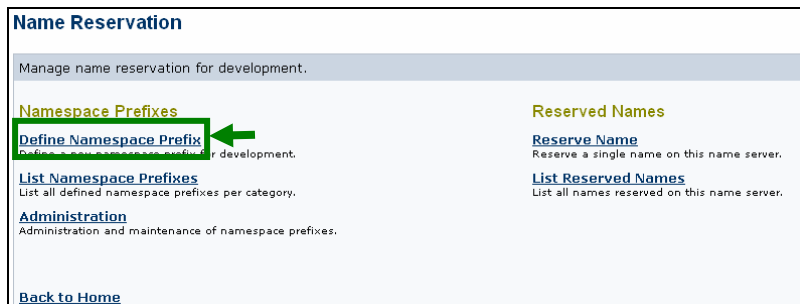
In this section, you will define the Namespace prefix for your development components.

2.3.3 Steps

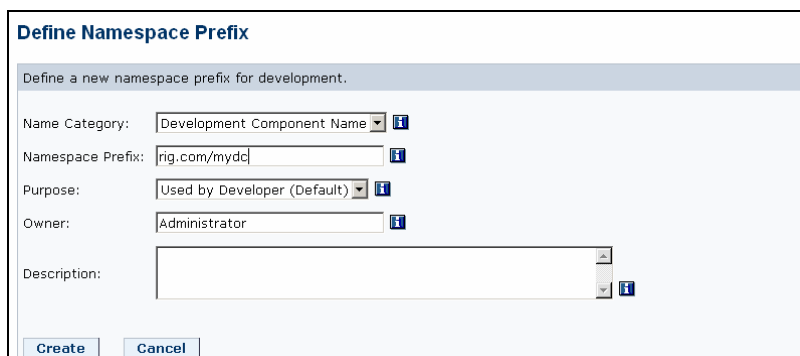
Choose the “Name Reservation” link in Development.



Choose the “Define Namespace Prefix” link under the Namespace Prefixes.

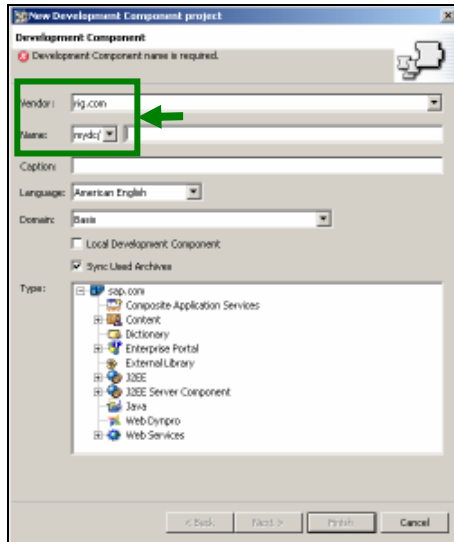


Select the Name Category: *Development Component Name* and choose **Create**.





The Namespace Prefix must be entered as **<vendor>/<prefix>**. You will see this prefix again later when creating the Development Component in the SAP Developer Studio (as figure below e.g. rig.com/mydc)



Name Category

Development Component Name: Names of the smallest reusable software bundles (development components). Hierarchical name separated by a slash (/). The file path in the Design Time Repository (DTR) is preceded by a vendor path that belongs to the development component name.

See SAP Help (<http://help.sap.com>) for detailed descriptions of the categories. [Path: SAP Library -> SAP NetWeaver -> Application Platform -> Java Technology in SAP Web Application Server -> Administration Manual -> Server Administration -> SAP System Landscape Directory -> Name Reservation -> Defining a Namespace Prefix -> Object Name Categories]

2.4 How to Create a Domain



2.4.1 Role

Administrator (Assign the user to the “JDI.Administrators” group)

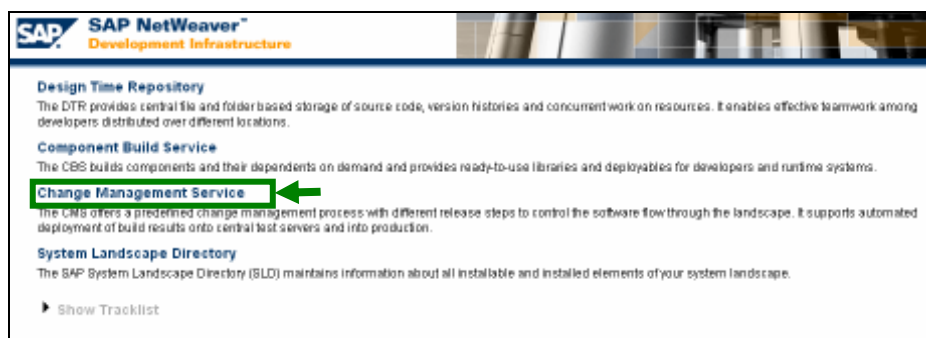
* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.

2.4.2 Goal

Here, you will create a domain in the Change Management Service (CMS). This step is necessary to enable development in SAP NetWeaver Developer Studio and to define the transport environment.

2.4.3 Steps

Open the browser and start the initial page of the SAP NetWeaver Java Development Infrastructure (http://<host>:<port>/devinf). Choose the *Change Management Service* link and the logon.



Users must be assigned to the following roles in advance using the *User Management tool*

(http://<hostname>:<port>/useradmin/).



- JDI.Administrator**
- JDI.Developer**

Assign a group to the aforementioned roles and add the users to this group.

* See the Appendix for detailed steps.

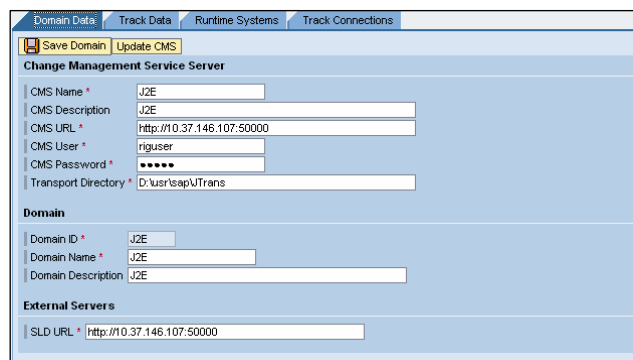


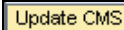
Messages informing you of the results of the action in question are displayed in the lower left-hand corner of the screen.

Switch to the *Landscape Configurator* screen and move to the “Domain Data” tab. Make the entries for the following parameters and save by choosing . Choose  to update the information from the SLD.

Messages informing you of results of the action in question are displayed in the lower left-hand corner of the screen.

- CMS Name
- CMS Description
- CMS
URL: http://<host name>:<port>
- CMS User
- CMS Password
- Transport Directory
- Domain ID
- Domain Name
- Domain Description
- SLD
URL: http://<host name>:<port>




Subsequently, whenever you create a new Software Component in the SLD, you have to go to this “Domain Data” tab and download the information by choosing .

2.5 How to Create a Track



2.5.1 Role

Administrator (Assign the user to the “JDI.Administrators” group)

* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.

2.5.2 Goal

Here, you will create a Track to define the transport environment. As the result, the developed objects can be transported through the systems (development, consolidation, test, and production system). The CMS also generates workspaces for each SC to be developed in the DTR and CBS buildspaces for these “logical” systems. In this track example, the runtime systems for DEV, CONS, TEST, and PROD will be maintained.

2.5.3 Steps

Switch to the “Track Data” tab in the Landscape Configurator (SAP NetWeaver Java Development Infrastructure (<http://<host>:<port>/devinf>)). *Change Management Service -> Landscape Configurator* and make the entries for General Track Data.

- Repository Type: Select **DTR**
 - * XI (Exchange Infrastructure track): This is used to transport design objects from various Integration Repositories or configuration objects from various Integration Directories.
- Track ID
- Track Name
- Track Description
- DTR URL: http://<host name>:<port>/dtr
- CBS URL: http://<host name>:<port>
- Development Configuration Path:



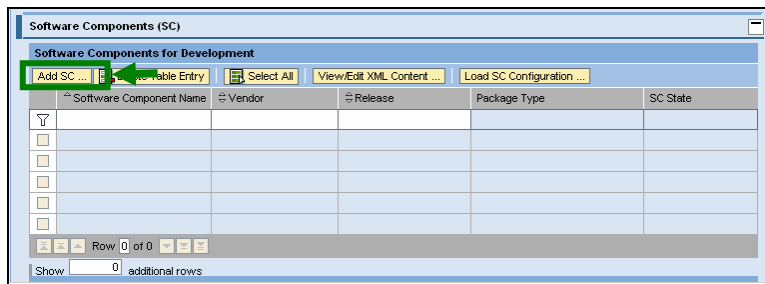
Development Configuration Path

This will be the hierarchical path used to structure the development configurations when they are downloaded in SAP NetWeaver Developer Studio. Example: You will see the hierarchy in the Developer Studio as the graphic below when you enter the Development Configuration Path “MYTRACK/1.0”.

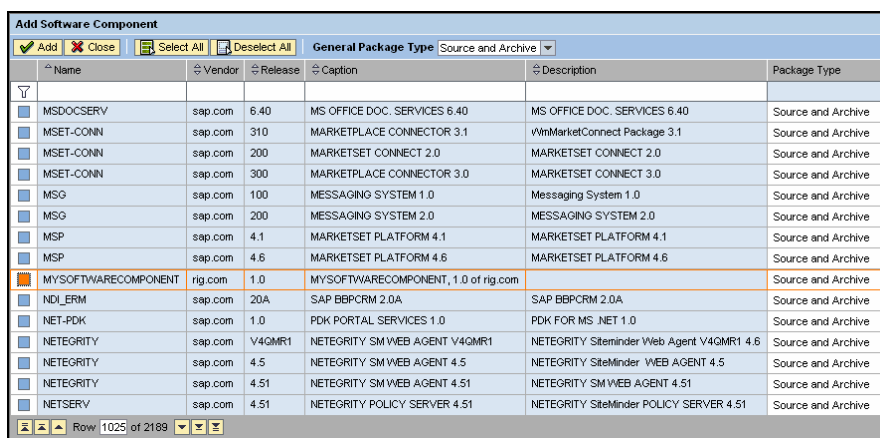
Development Configuration	Description	Internal Name
MYTRACK		
1.0		
MYTRACK_cons	Track for My Software Component_cons	3ZE_MYTRACK_C
MYTRACK_dev	Track for My Software Component_dev	3ZE_MYTRACK_D

Add the Software Component that you created in the SLD in section 2.2. Choose

Add SC ...



Select the relevant Software Component from the list, choose **Add** and close the popup.



If the Software Component is not displayed in the list, even though you created it earlier, go to the “Domain Data” tab and download the information by choosing **Update CMS**.



You can use the filter function to search the target software component by entering the value in the top row.



Disable automatic deployment

You can select the “Disable automatic deployment” function for the Development and Consolidation systems. By doing this, you remove the link between automatic deployment and activation in SAP NetWeaver Developer Studio. You then have to trigger the deployment manually. See SAP Note 770522 for details of manual deployment. We recommend that you keep the default setting.

Where-Used Check

You can configure up to four runtime systems in a development track. Additional productive systems can be added by using the track connections. The assignment to the system roles *Development*, *Consolidation*, *Test*, and *Production* determines the role of the runtime system. To avoid giving a different system role to a runtime system in another track or tracks, you can use the *Where-Used Check* function to see whether a runtime system is already being used in another track.

You can now see your newly created track at the left hand side of the browser.

Name	Description
MYTRACK	Track for My Soft...

2.6 How to Create Track Connections (Optional)

You can skip this section if you are working in a single track. You can only work through these steps if you have already defined the second track. (for example MYTRACKSP01)



2.6.1 Role

Administrator (Assign the user to the “JDI.Administrators” group)

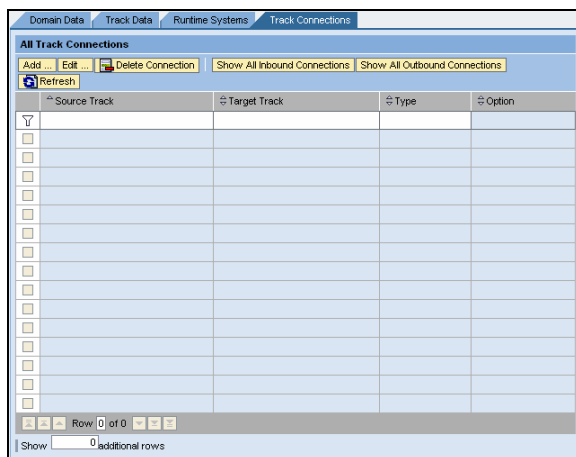
* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.

2.6.2 Goal

If you have layered development or maintenance developments, you will need to define multiple tracks. It will also be necessary to define connections between the tracks. Here, you will define these connections.

2.6.3 Step

Go to the “Track Connections” tab in the Landscape Configurator (SAP NetWeaver Java Development Infrastructure (<http://<host>:<port>/devinf>). *Change Management Service - > Landscape Configurator*) and choose **Add ...**.



A popup screen will appear where you can define the connections. Select the source and target track and define the connection type. Choose **Save** and then **Close** the popup.

Track Connection

Save Close

Connection

Source track: MYTRACK Track for My Software Component
 Target track: MYTRACKSP01 Track for My Software Component SP01

Connection Type

Transport
 Repair

Options

Transport all software components
 Transport developed software components
 Transport required software components
 Transport selected software components

Select All Deselect All

Vendor	Name	Type
rig.com	MYSOFTWARECOMPONENT	Developed
sap.com	SAP-IEE	Required
sap.com	SAP_BUILD	Required
sap.com	SAP_ITECHS	Required

Row 1 of 4

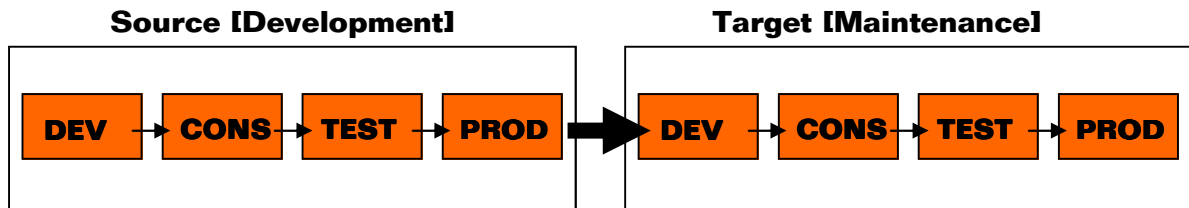


Note

Connection Type

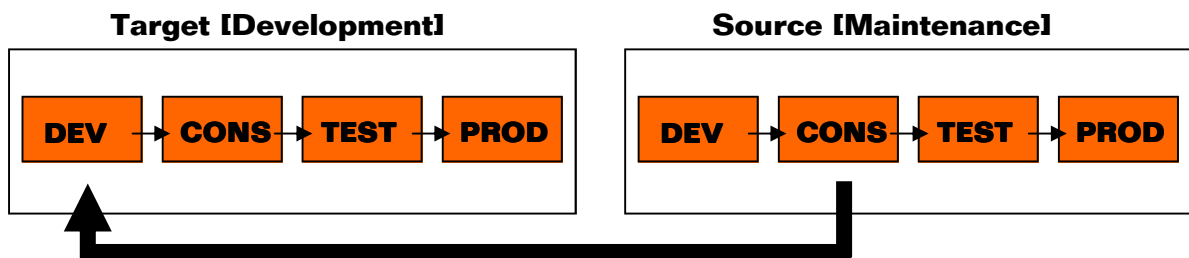
Transport

The approved development state of the development track (source) will be passed on to the maintenance track (target)



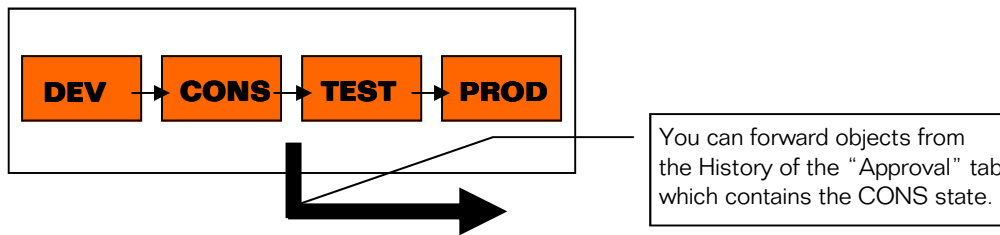
Repair

When the component in the DEV workspace of the maintenance track (source) is transported, it will also be transferred to the CONS workspace (maintenance track) queue and to the queue of the DEV workspace for the Development track (target).



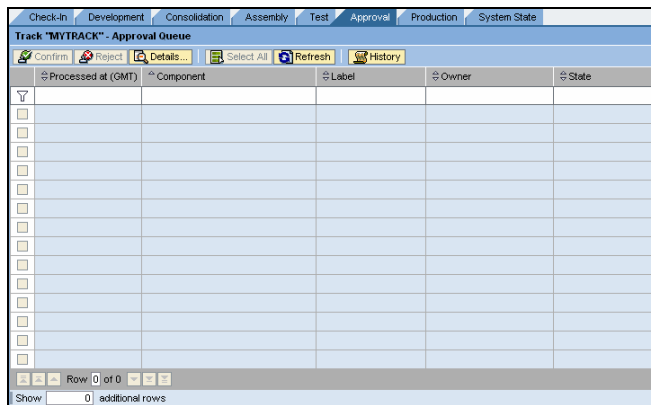
☑ **Forward a Component to a Specific Track**

You can forward a component to a track for which you have not maintained the connection in advance.

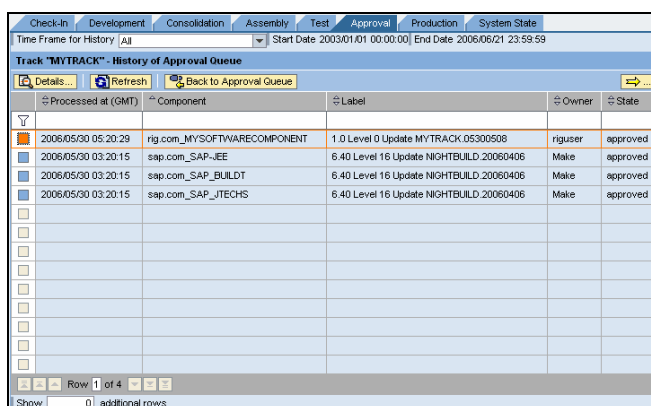


Below is the step to forward the component to the development system to a certain track.

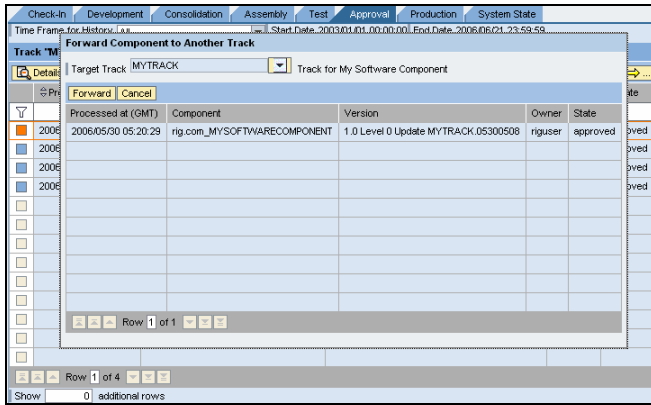
Go to the *Transport Studio* of the Change Management Service and select the "Approval" tab. Choose History.



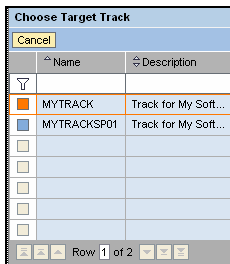
Select the target component that you want to forward and choose (Forward to track).



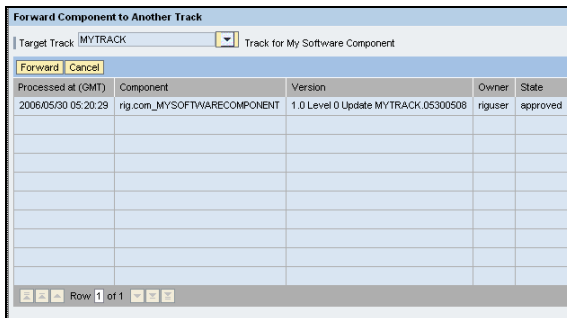
A popup screen will appear where you can define the target track when you choose the select icon ().



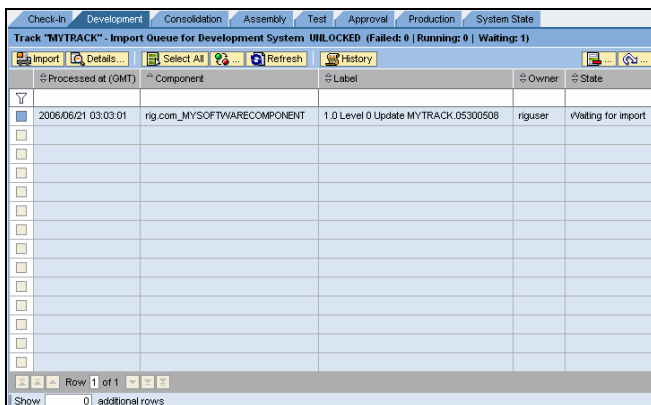
Choose the target track (where you want to forward the component to) from the list.



Choose **Forward** to transport the component to the DEV system of the target track.



In the target track's Development tab, you will now see the component with the status "Waiting for import".



2.7 How to Check In and Import Required Software Components



2.7.1 Role

Administrator (Assign the user to the “JDI.Administrators” group)

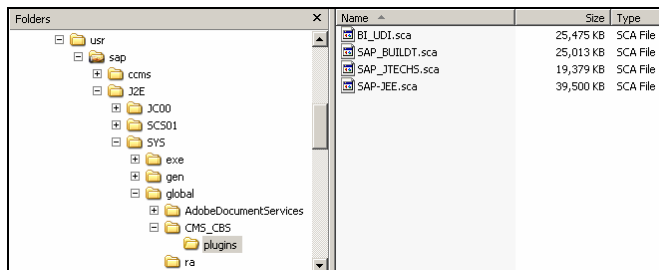
* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.

2.7.2 Goal

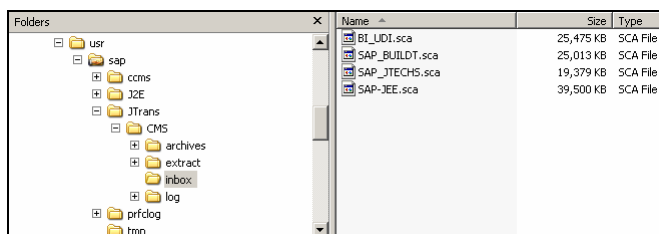
Here, you will check the software components into the track. This check-in is necessary to make the required software components available and to enable the build process with Central Build Service (CBS). You can also check in an SC for development if you want to change an existing SC.


2.7.3 Steps

Go to the directory where the archives (SAP_BUILD.T.sca, SAP_JTECHS.sca, and SAP-JEE.sca) are located. (<drive>:\usr\sap\<SID>\SYS\global\CMS_CBS\plugins)



Copy the files to the relevant Transport directory <drive>:\usr\sap\JTrans\CMS\inbox. The Transport directory is defined in the Landscape Configurator’s “Domain Data” tab.



Select the “Check-In” tab in the Transport Studio. You will now be able to see the three archive files in the table. Select all archive files by choosing  **Select All** and then choose **Check-In**.

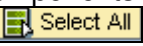

File Name	Component	Release	SP Number	PatchName	Version
SAP-JEE.sca	sap.com_SAP-JEE	6.40	16	NIGHTBUILD.20060406	NIGHTLYBUILD.20060406060400
SAP_BUILD.sca	sap.com_SAP_BUILD	6.40	16	NIGHTBUILD.20060406	NIGHTLYBUILD.20060406055600
SAP_JTECHS.sca	sap.com_SAP_JTECHS	6.40	16	NIGHTBUILD.20060406	NIGHTLYBUILD.20060406060000

 **Attention**

The manual step to copy the archives needs to be performed after the initial system set-up or after each upgrade.

* You can switch to the Transport Studio by choosing the “Transport Studio” link in the upper part of the screen.



Move to the “Development” tab and you will see the three software components with the “Waiting for import” status. Select all three components by choosing  and then choose .

Processed at (GMT)	Component	Label	Owner	State
2006/05/24 08:12:49	sap.com_SAP-JEE	6.40 Level 16 Update NIGHTBUILD.20060222	Make	Waiting for import
2006/05/24 08:12:53	sap.com_SAP_BUILD	6.40 Level 16 Update NIGHTBUILD.20060222	Make	Waiting for import
2006/05/24 08:12:56	sap.com_SAP_JTECHS	6.40 Level 16 Update NIGHTBUILD.20060222	Make	Waiting for import

3 Development – Working with the Component Model



3.1 How to Import the Development Configuration from SLD



3.1.1 Role


Developer (Assign the user to the “JDI.Developers” group)

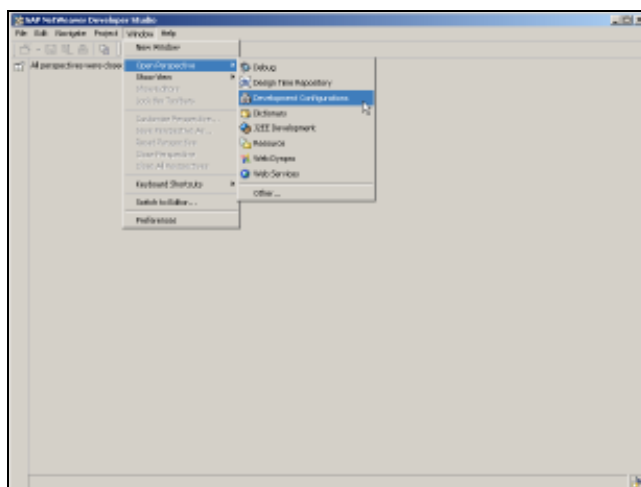
* The “JDI.Developer” role must be assigned to the “JDI.Developers” group.

3.1.2 Goal

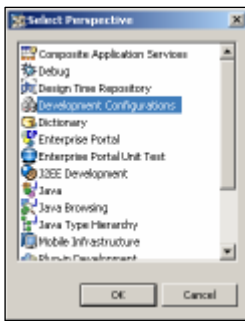
Here, you will establish the connection to the central development infrastructure. By importing the previously completed development configuration, you will be able to access all resources relevant for your work. You will be able to start the development in the SAP NetWeaver Developer Studio after importing the configuration.

3.1.3 Steps

Start the SAP NetWeaver Developer Studio and open the Development Configurations  perspective. Now choose *Window->Open Perspective->Development Configuration*

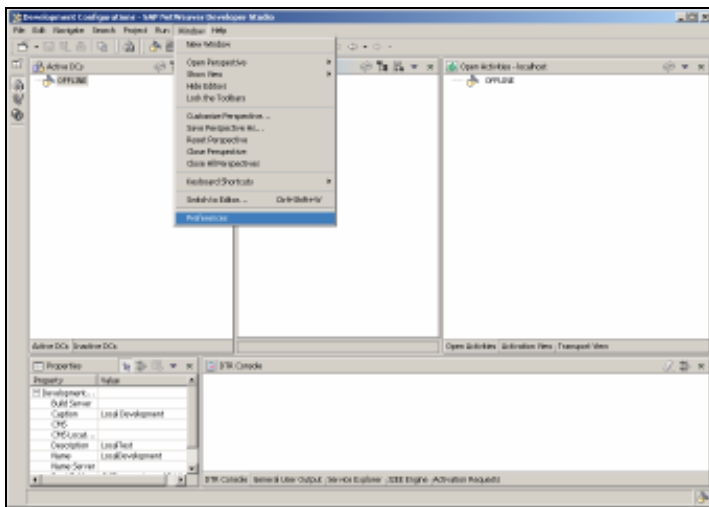


If you cannot find the Development Configurations, go to “Other...” and select the Development Configuration from the popup list.

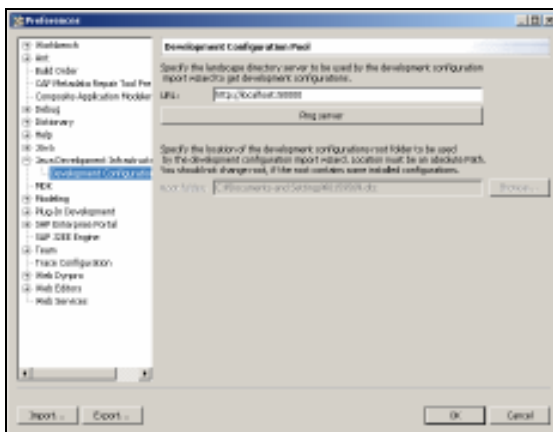


Another method is to right click on the tool bar and select “Customize Perspective ...”. Choose “Development Configurations” from the “Window > Perspective” menu.

You now need to establish the connection between the SAP NetWeaver Developer Studio and the SLD server. To do this, choose *Window->Preferences*



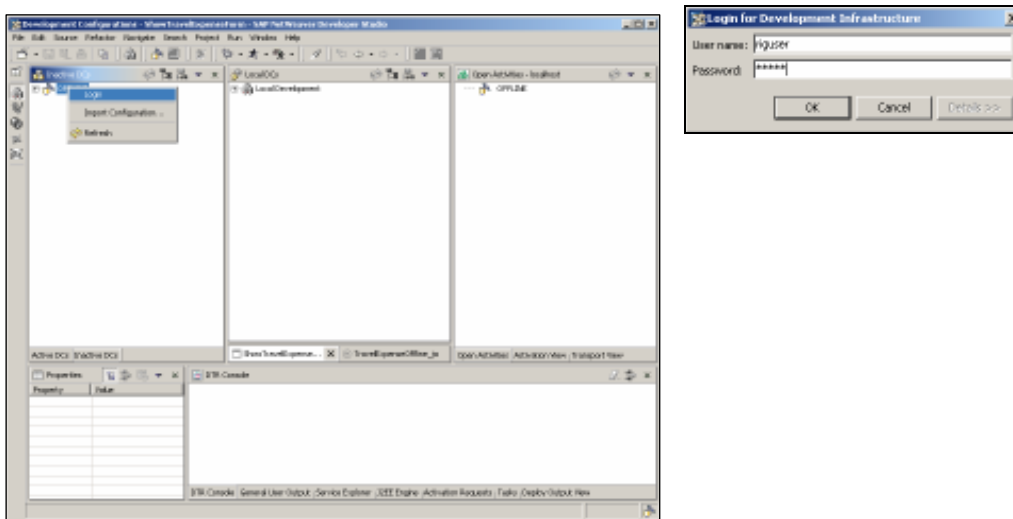
Choose *Java Development Infrastructure->Development Configuration Pool* and define the URL of the SLD server (<http://<host name>:<port>>). Choose the “Ping server” button to check whether the connection has been successfully established.





IF the ping fails, check whether the user is assigned to the appropriate role (JDI.Developer). See the Appendix for role assignment instructions.

Log on to the SAP NetWeaver Development Infrastructure by choosing Login from the context menu.

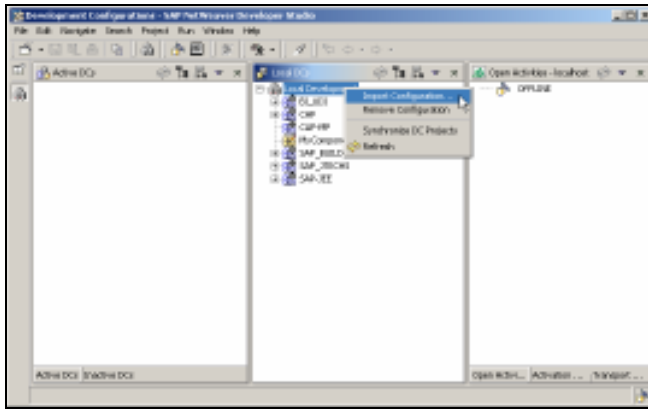


The login user must be assigned to the appropriate role (JDI.Developer).

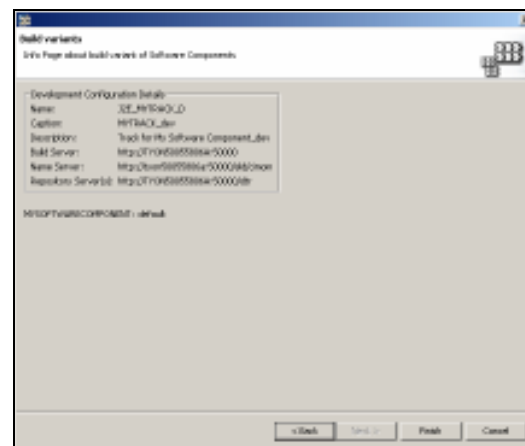
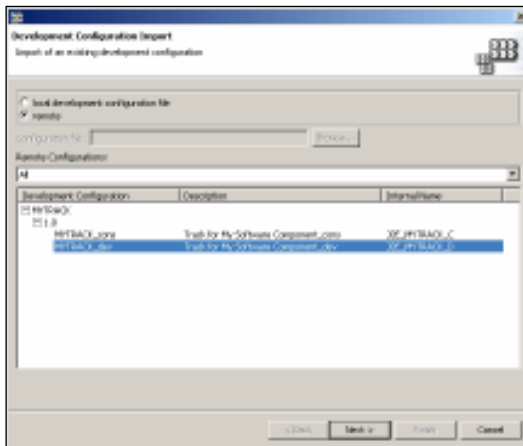
* See the Appendix for role assignment instructions.

The login user here is the user for development and does not necessarily have to be identical with the Administrator user.

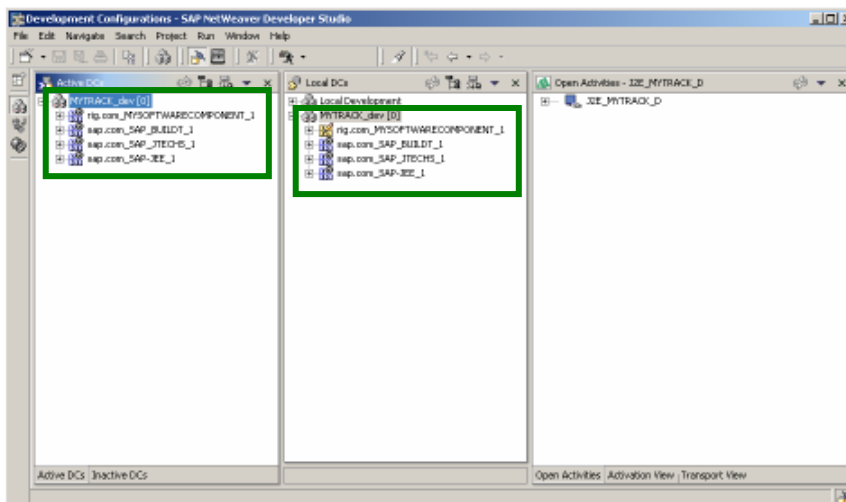
Select “Import Configuration...” from the context menu to import the configuration that you performed in section 2. Configurations stored in the CMS will be imported via connecting to SLD.



Select the “remote” option, choose the development track that was configured in section 2.5 and choose “Next”. Check the details in the next screen and choose “Finish”.



You will now see the imported development configuration in the Local DCs, Active DCs, and the Inactive DCs views. You are now able to start developing.



3.2 Developing in SAP NetWeaver Developer Studio



3.2.1 Role

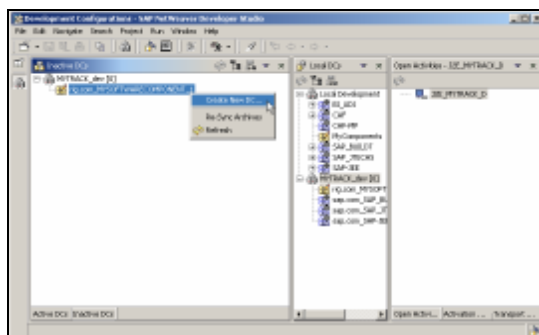
- Administrator (Assign the user to the “JDI.Administrators” group)
 - * The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.
 - * The activity naming convention should be defined by the administrator.
- Developer (Assign the user to the “JDI.Developers” group)
 - * The “JDI.Developer” role must be assigned to the “JDI.Developers” group.

3.2.2 Goal

You can start the development in the SAP NetWeaver Developer Studio after importing of the development configuration in section 2. The local development/changes will be included in an *Activity*. An *Activity* tracks modifications performed in the workspace and isolates the changes from other developers until the activity is “published”. It can be also used to group related changes forming one enhancement or fix together. This section describes the steps for creating the activity but will not provide details of development processes in the SAP NetWeaver Developer Studio.

3.2.3 Steps

Let us suppose you want to create a simple Web Dynpro Application. Go to the Inactive DCs view (changes can only be made in the Inactive DCs view and are not possible in the Active DCs view) and choose “Create New DC...” from the context menu.



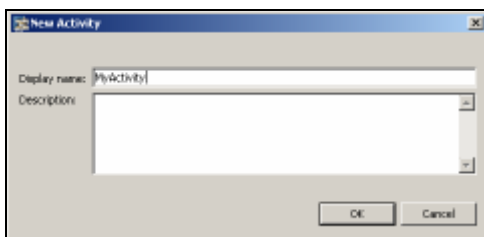
Make entries for *Vendor*, *Name* and *Caption* and choose Web Dynpro as the *type*. Choose Next to continue.



Choose the “New Activity...” button to create an *activity*. If an open *activity* is available, you can choose the existing one from the list.



Enter the name of the *activity* and choose “OK” to finish creating the new *activity*.



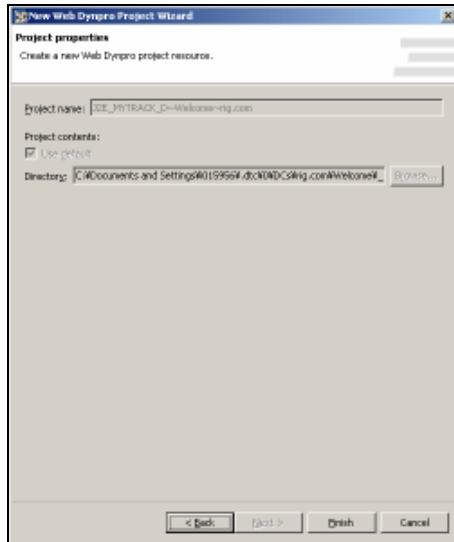
Attention

The *activity* name defined here will be displayed in the Transport Studio. It might therefore be useful to choose a name which helps you to identify files, version, etc. that are included in the activity. A GUID will be assigned automatically to make each activity globally unique.

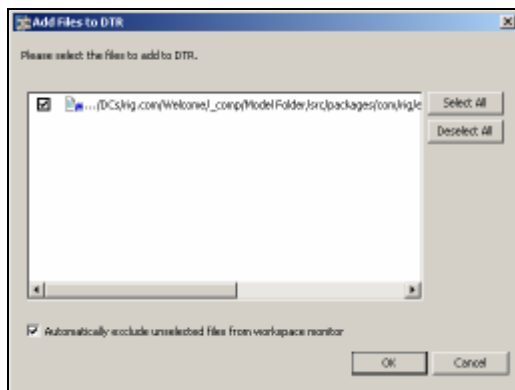
You will now see the created *activity* in the list. Select the activity and choose “Next”.



Choose “Finish” to include the change in the created *activity* in the above step.



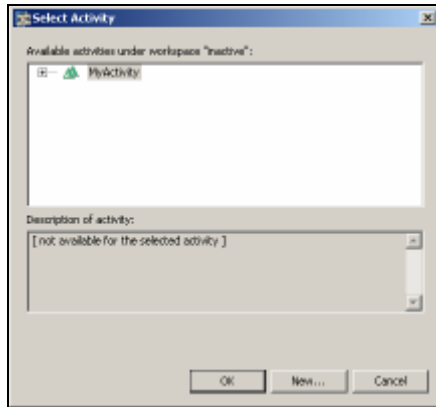
A Popup will appear asking whether the file should be added to the DTR whenever you make a change to the development objects. Choose “OK” to add include the changes in the activity.



The Developer Studio automatically calculates correctly which files should be stored and prompts you to add these to the DTR. You should therefore normally accept the addition. If an error occurs, choose Preferences -> Team -> Ignored resources and check whether all file types are correctly set.

Add the change to the *activity* that you created above. Select the relevant *activity* from the list and choose “OK”.

* The last used activity will be highlighted.



3.3 How to Create Public Parts of a DC



3.3.1 Role

Developer (Assign the user to the “JDI.Developers” group)

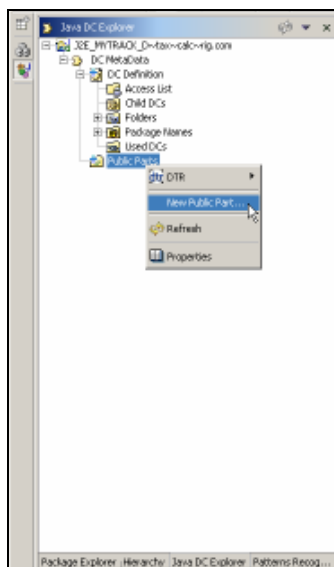
* The “JDI.Developer” role must be assigned to the “JDI.Developers” group.

3.3.2 Goal

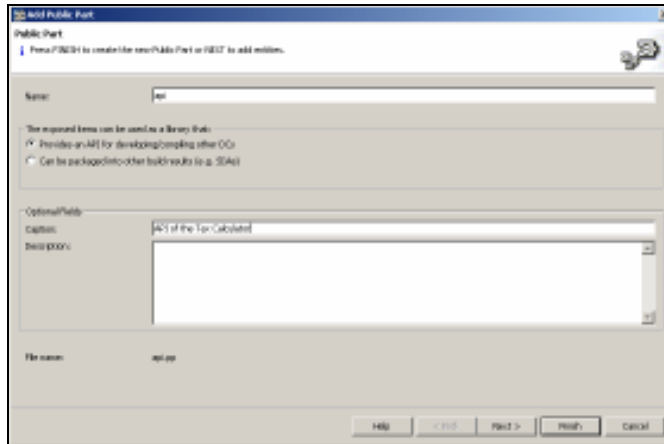
Public parts are the interfaces of a component. They consist of a list of development objects offered to other components.

3.3.3 Steps

Open the relevant Explorer view which you can find under the DC MetaData node (The example below shows the Java DC Explorer view), and select the Public Parts node. Choose “New Public Part...” from the context menu.



Enter the name of the public part in the dialog and determine the type from the radio button. Choose “Next” to edit the entity.



Note

Usage type

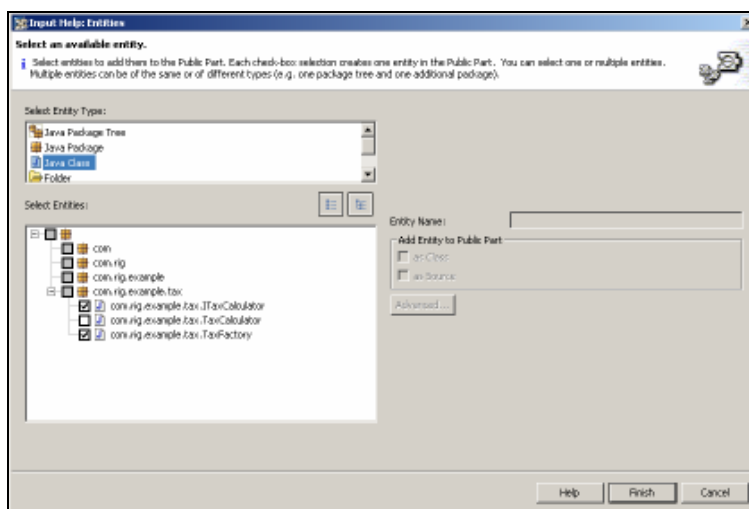
- Provide an API for developing/compiling other DCs

The objects (symbols, classes, etc.) in the public part of a component are needed to compile the other component. Add only those objects that you want to be visible on the outside, such as interfaces, but never their implementations.

- Can be packages in other build results (e.g. SDA)

The objects (symbols, classes, etc.) in the public part of a component are assembled to a larger unit by the other component, such as an archive or an application. Usually, the public part also contains some private objects of their component, for example, the interfaces and their implementation. You cannot use a public part of this type to compile component, or vice versa. However, you can assign two public parts of different types to a component, where both contain the same development objects. This is especially needed if a DC (such as Java DCs) has no deployable result.

Select the Entity Type and the relevant entities from the hierarchical display and choose “Finish”.



3.4 How to Use a Public Part (Defining Usage Dependency)

3.4.1 Role

Developer (Assign the User to the “JDI.Developers” group)

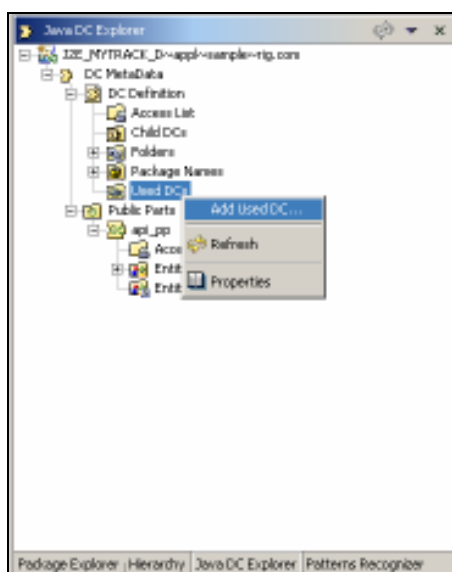
* The “JDI.Developer” role must be assigned to the “JDI.Developers” group.

3.4.2 Goal

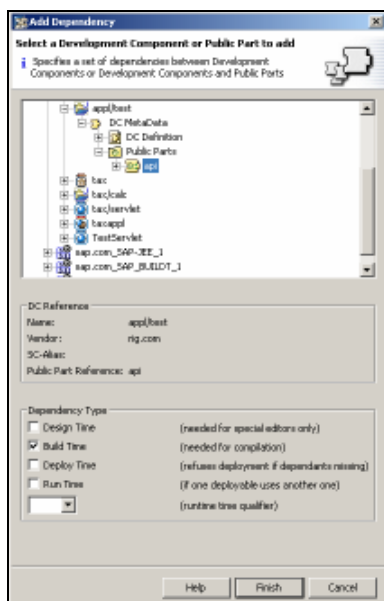
The DC can use the public part once the declaration of the use access has been established. Here, you will define the usage dependency to the public part.

3.4.3 Steps

Open the relevant Explorer view that you can find under the DC MetaData node (The example below is the Java DC Explorer view), and select the “Used DCs” node of the relevant DC MetaData. Choose “Add Used DC...” from the context menu.



Select the Public Parts target from the hierarchy, define the “Dependency Type” and choose “Finish” to complete the declaration.





- Design Time (needed for special editors only)
Currently not used
- Build Time (needed for compilation)
This dependency specifies that a component is used during compilation or when wrapping another component. The DC will require the build results from the added DC (Public Part).
- Deploy Time (refuses deployment if dependants missing)
This dependency specifies that a used component must already exist in the runtime system. An error message will be displayed during the build process if the added DC (Public Part) is not deployed.
- Runtime (if only deployable uses another one)
This dependency specifies that a component requires another component at runtime. It therefore only makes sense to use it if both DCs have deployable results.

Refer to Note 798872 for details of DC dependencies.

3.5 How to Check In and Activate



3.5.1 Role

Developer (Assign the user to the “JDI.Developers” group)



* The “JDI.Developer” role must be assigned to the “JDI.Developers” group.

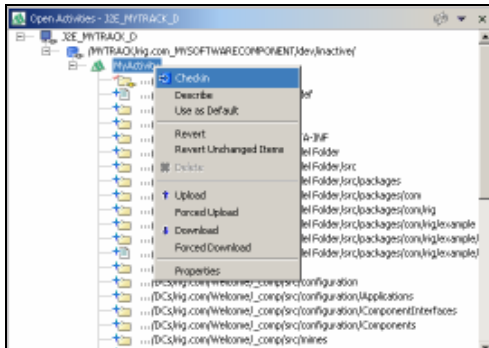
3.5.2 Goal

After completing the unit test in the local environment, you can now check the changes into the workspace. After the changes are checked in to the workspace, you need to perform the activation. Activation triggers the build process of your DCs in the Component Build Service (CBS). If the runtime system is configured for the DEV system, the automatic deployment is also triggered after the successful build. The changes will then be available to other developers. Using activated DCs guarantees that they are in sync with the latest archive states stored in the CBS.

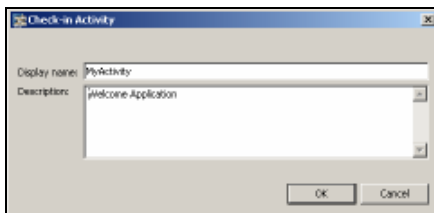


3.5.3 Steps

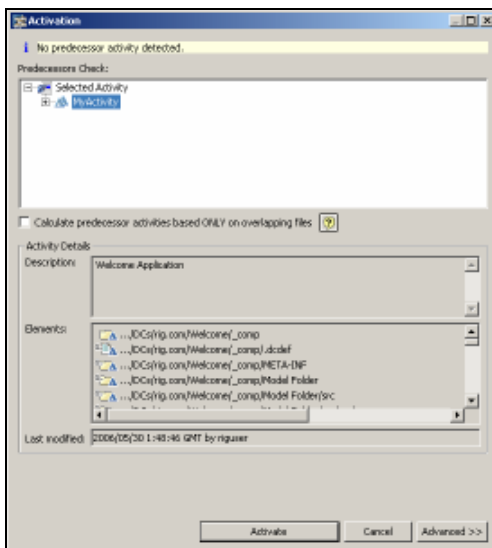
Switch to the Development Configuration  perspective and choose the “Open Activities” view. Here, you will see the activity including the change list. Select the activity and choose Checkin  from the context menu.



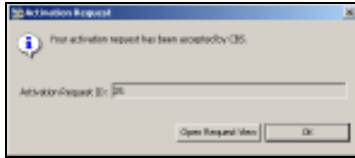
Add the description (optional) to the *activity* and choose OK.



Select the corresponding *Activity* and choose “Activate”.



Choose “OK” or “Open Request View” to start the activation. The view showing the build results will be opened too if you choose “Open Request View”.



3.6 Conflict Resolution

3.6.1 Role

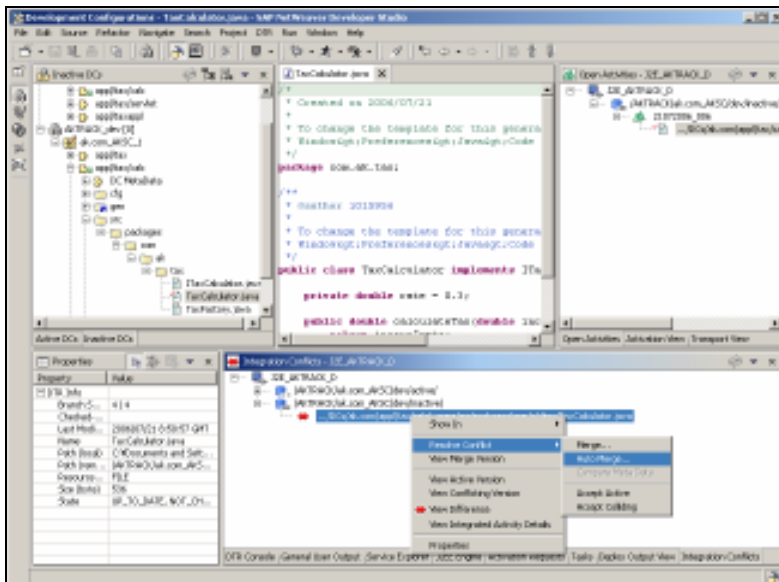
Developer (Assign the user to the “JDI.Developers” group)

3.6.2 Goal

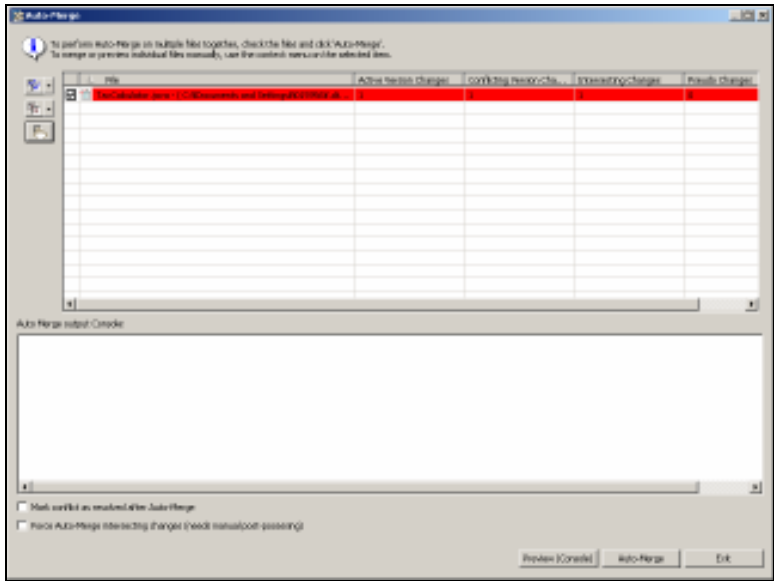
Source code version conflicts can occur, for example, if development is performed with multiple developers. The conflict between the two versions will be detected automatically, and you can resolve the conflict from the “Integration Conflict” view.

3.6.3 Steps

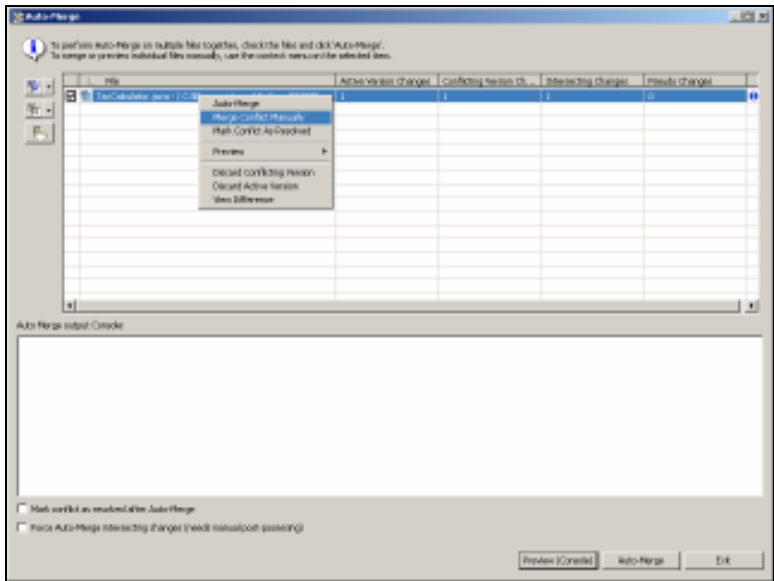
Open the “Integration Conflict” view to display where the version conflict occurred. Select “Auto Merge...” from the context menu.



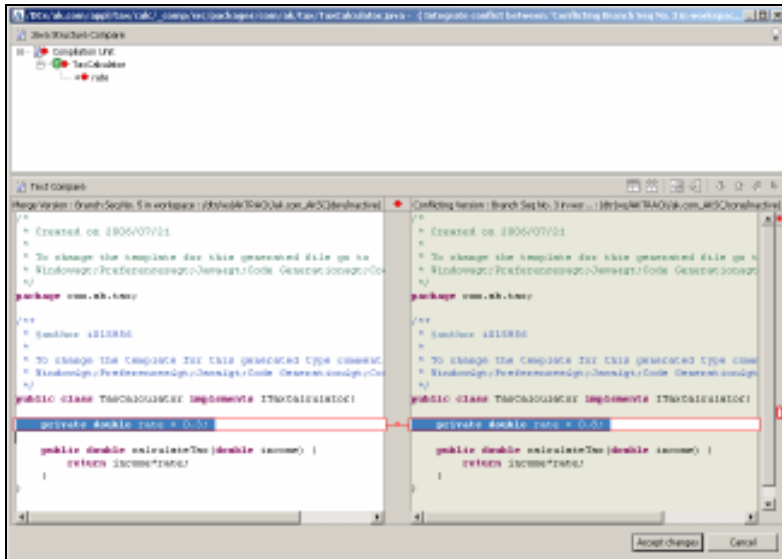
The list of the source code with conflicts will be displayed.



Select the relevant source code and choose “Merge Conflict Manually” from the context menu.



The two source code versions will be displayed and will indicate the conflict line. Choose “Accept changes” after resolving the conflict by merging the code lines.



3.7 How to Release Components



3.7.1 Role



Developer (Assign the user to the “JDI.Developers” group)

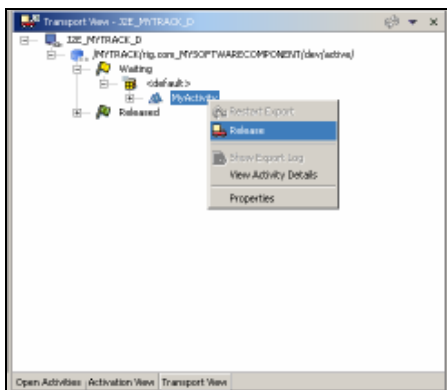
* The “JDI.Developer” role must be assigned to the “JDI.Developers” group.

3.7.2 Goal

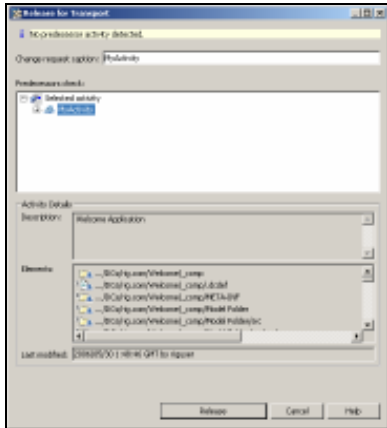
The development process will complete by releasing the activities in the SAP NetWeaver Developer Studio. As a result of releasing the activities, the changes will be exported and placed in the import queue of the consolidation system in the Change Management Service (CMS).

3.7.3 Steps

Switch to the “Transport View” and select the activity in the Waiting  state and choose Release  from the context menu.



Select the relevant activity and choose “Release”. You will now see the activity in the Release  state.



4 Transporting Components



4.1 How to Import to the Consolidation System



4.1.1 Role

Administrator (Assign the user to the “JDI.Administrators” group)

* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.


4.1.2 Goal

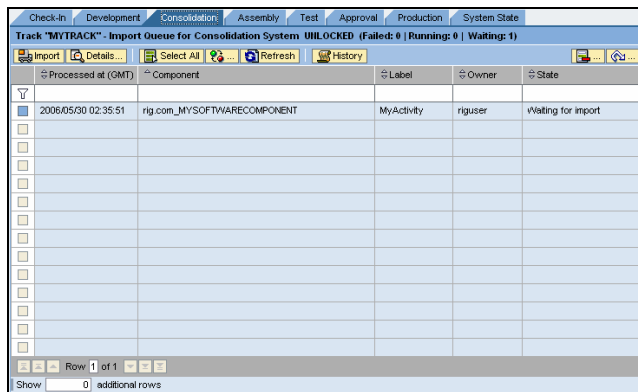
The system administrator imports the queue to the consolidation system. The released changes will be integrated into the DTR workspace of the consolidation system. The integration testing can be performed in this consolidation system, and bug fixing can take place here.

4.1.3 Steps

Open the browser and start the initial page of the SAP NetWeaver Java Development Infrastructure (<http://<host>:<port>/devinf>). Select the Change Management Service link and switch to the Transport Studio.



Switch to the “Consolidation” tab. Here, you will see the software component that was released from the SAP NetWeaver Developer Studio in section 3.5. Select this component and choose .



Processed at (GMT)	Component	Label	Owner	State
2006/05/30 02:35:51	rig.com_MYSOFTWARECOMPONENT	MyActivity	riguser	Waiting for import

4.2 How to Assemble



4.2.1 Role


Administrator (Assign the user to the “JDI.Administrators” group)

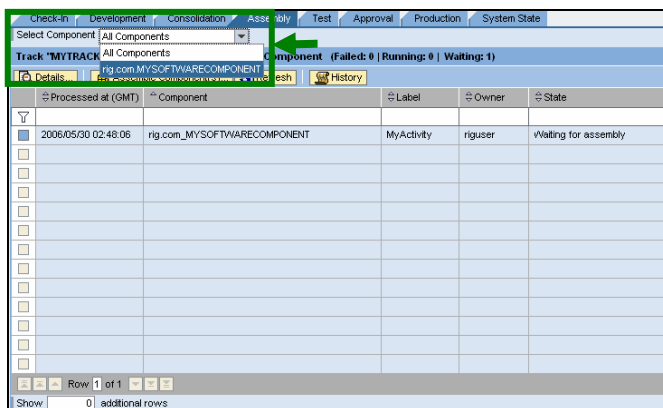
* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.

4.2.2 Goal

You will create a new software component archive (SCA) of the application from the previously consolidated state of the software changes. Normally, central tests are carried out in the consolidation state prior to the assembly step.

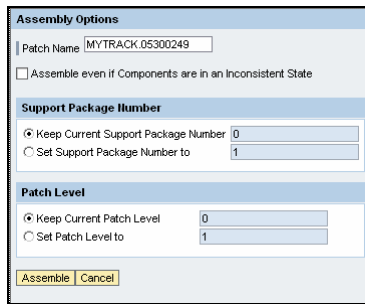
4.2.3 Steps

Open the browser and start the initial page of the SAP NetWeaver Java Development Infrastructure (<http://<host>:<port>/devinf>). Select the Change Management Service link and switch to the Transport Studio. Select the “Assembly” tab. You will now see the change request with the status “Waiting for Assembly”. Select your software component from the “Select Component” pull down list to assemble only your own software components. If “All Components” is selected, all software components defined in the track will be assembled. Choose  to trigger assembly.

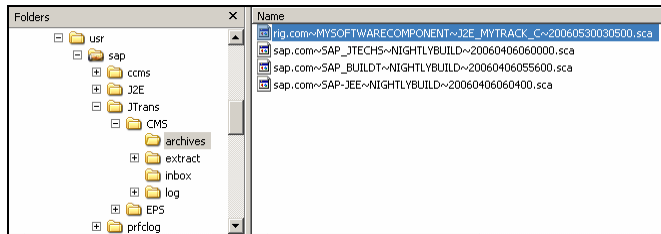


Processed at (GMT)	Component	Label	Owner	State
2006/05/30 02:48:06	rig.com_MYSOFTWARECOMPONENT	MyActivity	riguser	Waiting for assembly

Retain the default settings in the Assembly Option dialog box and choose Assemble.



CMS now creates a new software component archive (SCA) in the directory <drive>:\usr\sap\Jtrans\CMS\archives. The name of the SCA is structured as follows: <vendor>~<SC name>~<domain>_<track ID>_<system role>~<timestamp>.sca. If the test system is defined in the runtime system, the files can be imported into the test J2EE Engine.



4.3 How to Import the Archives to the Test System



4.3.1 Role

Administrator (Assign the user to the “JDI.Administrators” group)


* The “JDI.Administrator” role must be assigned to the “JDI.Administrators” group.

4.3.2 Goal

You import the software component archives (SCA) created in the assembly step into the central test system. The application will be deployed automatically, and testing can be performed.

4.3.3 Steps

Import the software component archive (SCA) you created in the assembly step into a central test system. The import also automatically deploys the application.

Select the “Test” tab (this tab appears if you have defined the “Test” runtime system in your track) and you will see the component in the “Waiting for Import” state. Select the component and choose  Import.

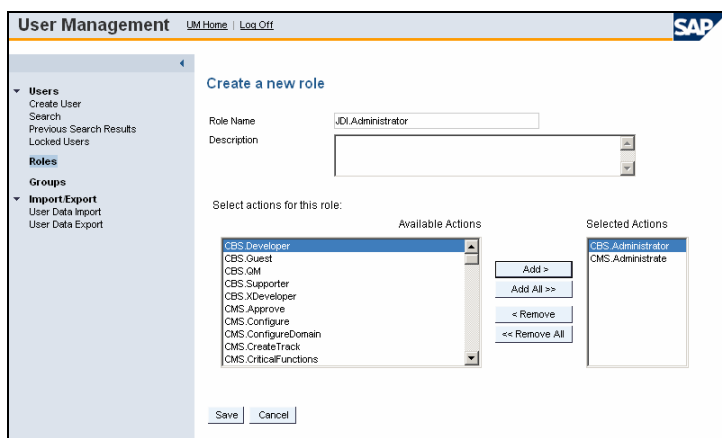
5 Appendix A: Steps for Creating a and Assigning it to a User

Below are the steps for manually creating a role/group and making the assignments. See Appendix B for information about how to use the Template Installer to automate the configuration steps.

5.1 Create a Role

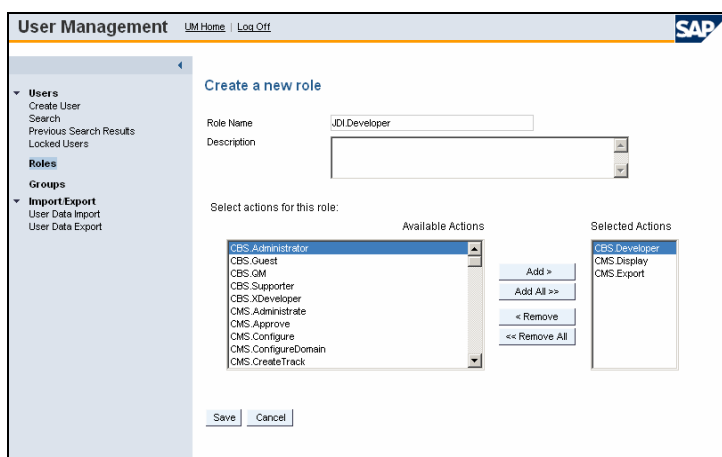
Open the SAP J2EE Engine User Management (<http://<hostname>:<port>/useradmin>), choose “Roles” from the menu and select “Create” to create a new role. Enter the role name “JDI.Administrator”, add the following actions to this role and save.

- CBS.Administrator
- CMS.Adminstrate



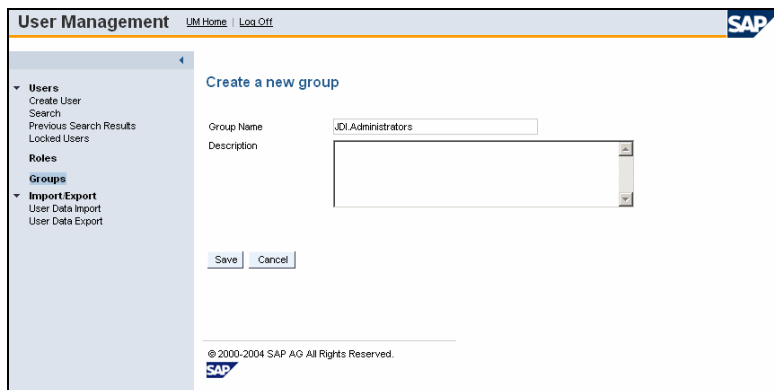
Repeat the same step to create the role “JDI.Developer” and add the following actions.

- CBS.Developer
- CMS.Display
- CMS.Export



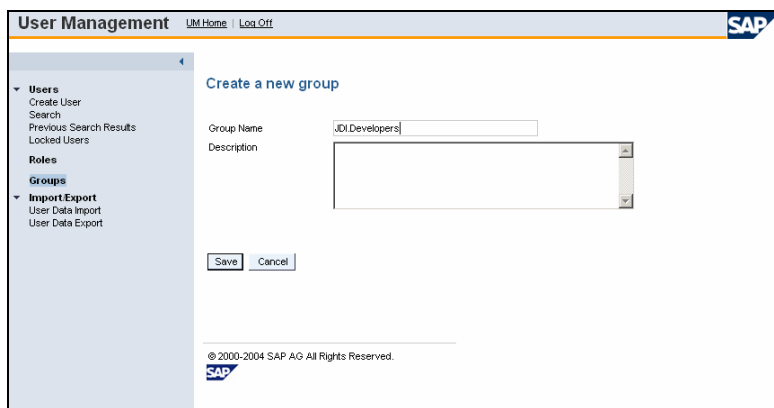
5.2 Create a Group

Choose “Groups” from the menu and choose Create in order to trigger the creation of a new group. Enter “JDI.Administrators” as the name of the group and save your entries.




The screenshot shows the SAP User Management interface. The left sidebar has a menu with 'Users', 'Roles', 'Groups', and 'Import/Export'. The 'Groups' section is active. The main area is titled 'Create a new group'. It contains a 'Group Name' field with 'JDI.Administrators' and an empty 'Description' text area. There are 'Save' and 'Cancel' buttons at the bottom. The footer shows '© 2000-2004 SAP AG All Rights Reserved.' and the SAP logo.

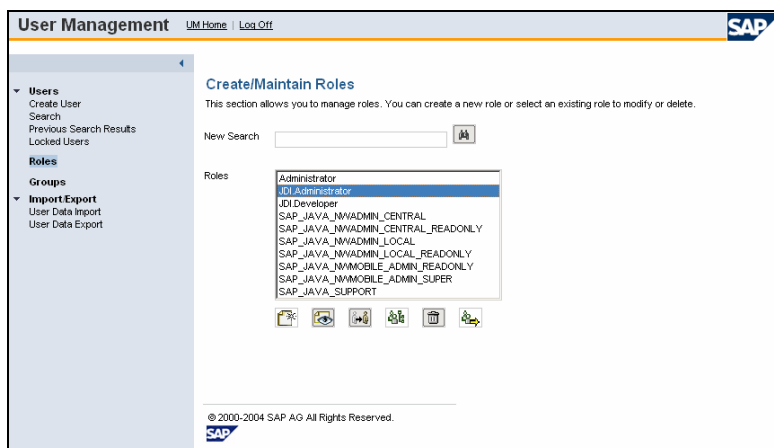
Repeat the step to create the “JDI.Developers” group.




This screenshot is identical to the previous one, but the 'Group Name' field contains 'JDI.Developers'.

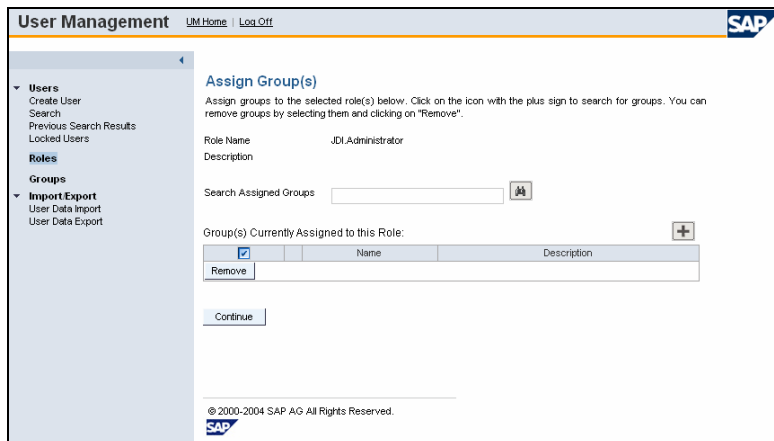
5.3 Assign a Group to a Role

Choose “Roles” from the menu. Select the “JDI.Administrator” role from the list and choose  (Assign Groups to ...).

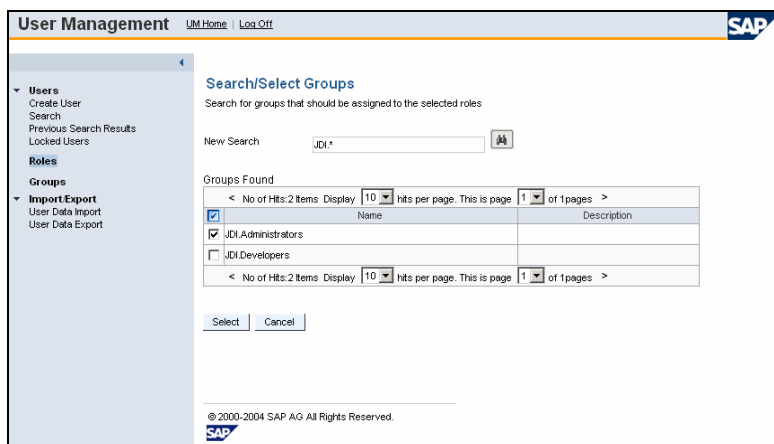


The screenshot shows the 'Create/Maintain Roles' screen. The left sidebar has 'Roles' selected. The main area has a 'New Search' field and a list of roles. The 'JDI.Administrator' role is highlighted. Below the list are several icons, including one with a plus sign and a group of people icon, which is the 'Assign Groups to ...' icon mentioned in the text. The footer shows '© 2000-2004 SAP AG All Rights Reserved.' and the SAP logo.

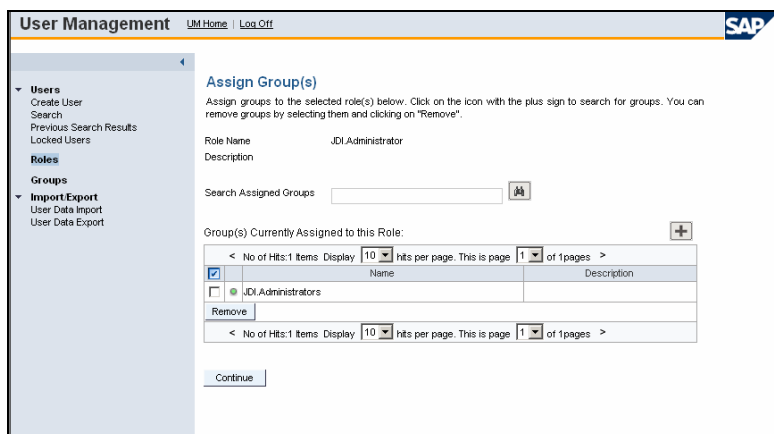
Choose  to assign a group.



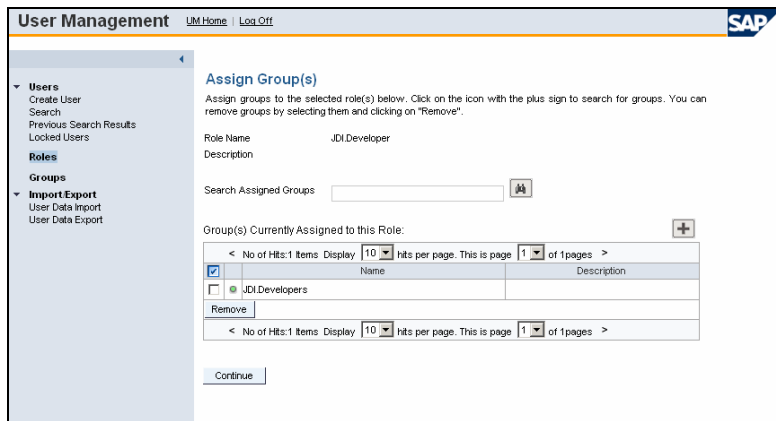
Search for the “JDI.Administrators” group. Select the checkbox and choose “Select”.



Choose “Continue” to complete the assignment.

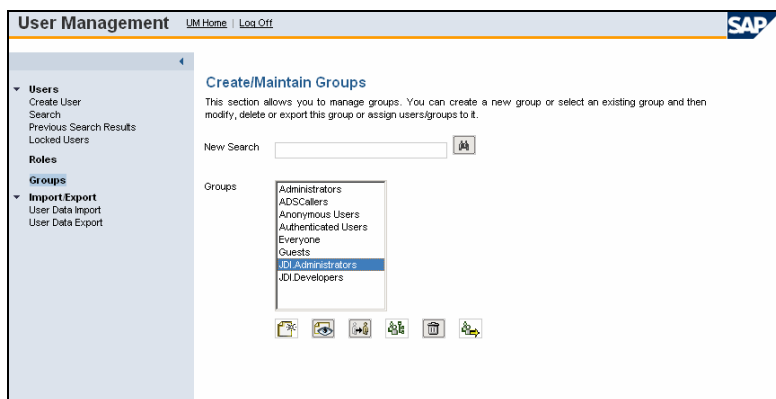


Repeat the step to assign the “JDI.Developer” role to the “JDI.Developers” group.

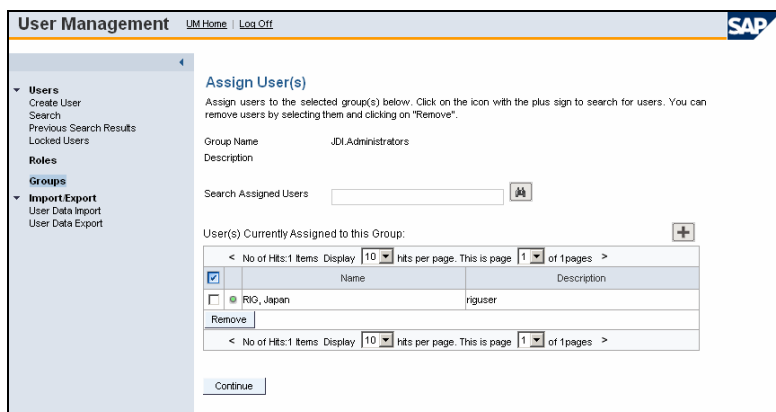


5.4 Assign a Group to a User


Choose "Groups" from the menu. Select the "JDI.Administrators" group to assign the administration authorization to a specific user. Choose (Assign Users to ...).



Choose , search for the target user and complete the assignment.



Repeat the step to add a development authorization (assign to the "JDI.Developers" group) to a specific user.


User Management [LM Home](#) | [Log Off](#) 


- Users
 - Create User
 - Search
 - Previous Search Results
 - Locked Users
- Roles
- Groups
- Import/Export
 - User Data Import
 - User Data Export

Assign User(s)

Assign users to the selected group(s) below. Click on the icon with the plus sign to search for users. You can remove users by selecting them and clicking on "Remove".

Group Name: JDI.Developers
Description:

Search Assigned Users: 

User(s) Currently Assigned to this Group: 

< No of Hits: 1 Items Display 10 hits per page. This is page 1 of 1 pages >	
<input checked="" type="checkbox"/>	Name Description
<input type="checkbox"/>	RIC, Japan /iguser
Remove	
< No of Hits: 1 Items Display 10 hits per page. This is page 1 of 1 pages >	

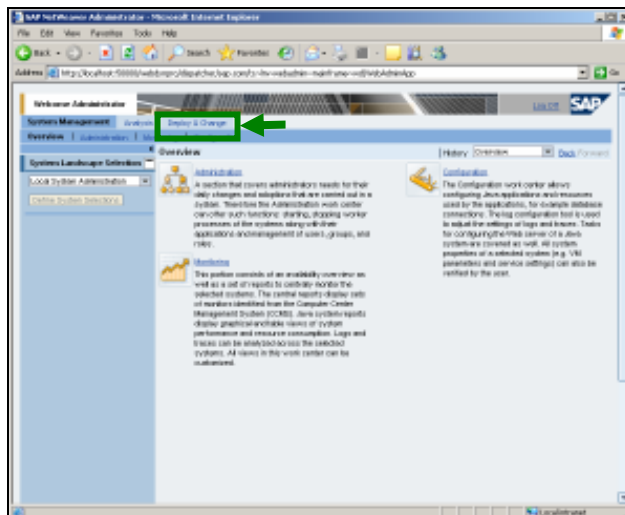
[Continue](#)

6 Appendix B: Template Installer

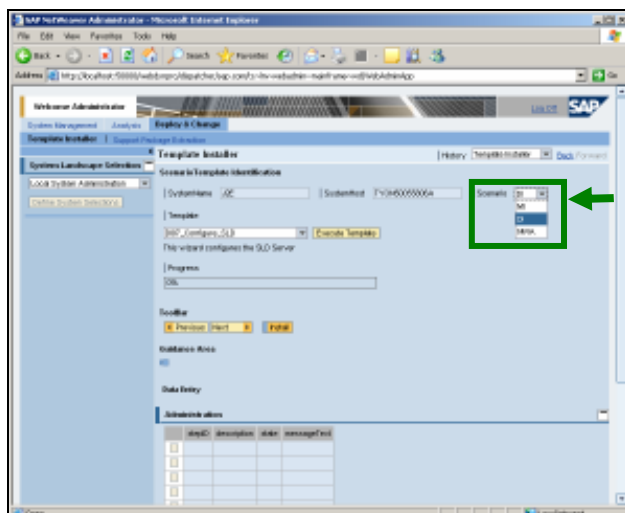
Template Installer can be used to set up the Development Infrastructure with all post-installation steps. Configurations such as the ones below can be carried out using the Template Installer. This will make it easier for you to start working with the Development Infrastructure. Note that this function can be used in SAP NetWeaver 04 SPS 14 and higher.

- Create User
- Create Roles
- Create Groups
- Role and Group assignments
- Create Tracks
-
-

Open the *NetWeaver Administrator* (<http://<hostname>:<port>/nwa>) and choose the “Deploy & Change” link.



Select the “DI” Scenario from the drop down list.



<http://www.sdn.sap.com/irj/sdn/howtoguides>