Table of Contents:

Introduction 3

Deinstallation: 6.30 SAP J2EE Engine and SAP NetWeaver Developer Studio (SP2) 3

Installation of Release 6.40 (SP4) J2EE Engine plus Developer Studio (NW04) 5
  Installation preparation 5
  Installation tasks 5

J2EE Project Migration 6
  Purpose 6
  Migration Process 6
    Extract information from the old projects 6
    Create new projects 7
    Handling of complex project structures 8
    Copy the contents of the old projects into the new projects 8
    Adjust references to other projects/libraries 10

Web Dynpro Project Migration 11
  Purpose 11
  Migration Process 11
  Developer Studio - Actions 13
  UI Elements - Actions 13
  Simple Types - Actions 14
  Deployment 14
  Configuration Aspects 15
    Developer Studio Configuration 15
    Web Dynpro Configuration 15
    Portal integration 15
    SLD Integration 15

General Restrictions 16

Troubleshooting 16

Further Information 16
  Documentation 16
  Migration Problems 17
Introduction

SAP Web Application Server Release 6.40 and the SAP NetWeaver Developer Studio are part of the development platform of the first SAP NetWeaver Release, 04.

An SAP Java server including an integrated development environment have been developed and provided to the application developers already with SAP AS 6.30, so there might be some development projects which are intended to run on the 6.40 platform of NW04, or for which development will continue on the new platform which have to be migrated to the 6.40 server.

This guide describes the procedures how to migrate Java applications from the 6.30 SAP Web Application Server platform (SP2) to the 6.40 server and development environment (SP4).

Both for the migration of J2EE projects and Web Dynpro projects, or combinations of both, the application developer has to execute certain steps to avoid problems running the Web applications on the new platform.

Since a J2EE project and a Web Dynpro project have to be handled in a different way during migration, you will following find two main chapters for each project type.

At very first, you have to consider whether you would like to migrate your applications on one physical machine, or not. Since the installation procedure is a little bit different in the second case, an early decision has advantages. But, of course, it is also possible that you start with the 6.40 migration on your machine first, and that you move each project to another server with a 6.40 installation lateron.

The first steps which are following described are how you deinstall your 6.30 version of the SAP J2EE Engine and the Developer Studio, and how you install the 6.40 version. These procedures are valid for both the 6.30 J2EE projects and the 6.30 Web Dynpro projects.

Deinstallation: 6.30 SAP J2EE Engine and SAP NetWeaver Developer Studio

If you would like to migrate your 6.30 projects to the 6.40 release on the same physical machine, you have to start with a complete deinstallation of the 6.30 version. That means, you uninstall the J2EE Engine plus the development environment, while your workspace (default c:\Documents and Settings\<SAP User Id>\Documents\SAP\workspace) you had defined for your SP2 project files will not be deleted but kept.

If you will migrate your projects to a 6.40 version of the J2EE Engine and the Developer Studio on another workplace where there is no installation of the engine and the Studio, you can skip this description and go on with the next chapter “Installation of the NW04 Developer Workplace”.

For the 6.30 deinstallation, do the following steps:

1. For deinstallation of the SAP J2EE Engine, goto your local directory c:\usr\sap\<SAP System Id>\JC\<J2EE Instance Number>, and double-click the UNINSTSAP.EXE.

2. For deinstallation of the SAP NetWeaver Developer Studio, goto the default installation directory c:\Program Files\SAP\JDT\eclipse\ in your file system, or choose your corresponding customized installation path, and start the uninstall by double-clicking the JDTsetup.exe file to start the Developer Studio Configuration Wizard.

3. Choose option Uninstall all SAP NetWeaver Developer Studio 2.0.4 components, and start the deinstallation process with Finish.
What kind of configuration task do you want to start?

- Update: Synchronize all SAP NetWeaver Developer Studio 2.0.4 components with the installation source
- Add or remove one or more SAP NetWeaver Developer Studio 2.0.4 components
- Uninstall all SAP NetWeaver Developer Studio 2.0.4 components
Installation of Release 6.40 J2EE Engine plus Developer Studio (NW04)

The NW04 Rampup version of the Developer Studio is developed on basis of the open source Eclipse version 2.1.2. The valid Eclipse version is part of the standard installation files as they are provided from SAP for customers and partners.

Installation preparation

Before you start with the installation, do the following steps to avoid problems during the installation process:

1. SAP Web Application Server Release 6.40 uses the SUN Java Development Kit (JDK) version 1.4.2 instead of JDK 1.3.1 which was the valid kit for 6.30.
   
   If you have already another JDK version installed on your host, you can install the required 1.4.2 version additionally. You do not need to uninstall the existing JDK, but you have to set the JAVA_HOME environment variable within your system settings (advanced):

   ![JAVA_HOME Environment Variable](image)

2. Check if the SAP Logon Group is closed. Otherwise, you will get a message during the installation process, and you will be asked to close this group.

3. Goto your local file system, and delete the complete .metadata directory from your workspace path, if you plan to migrate your project files to the same physical machine.

Installation tasks

For installation of the 6.40 version of the SAP NetWeaver 04 Workplace, read the installation instructions which are part of the SAP NetWeaver Administration Guide. Goto the SAP Service Marketplace [http://service.sap.com/instguides](http://service.sap.com/instguides).
J2EE Project Migration

Purpose

The main focus during migration of 6.30 J2EE development projects to the 6.40 platform is considering the restructuring of the J2EE projects since there are less J2EE project types as of 6.40 than with 6.30 to simplify J2EE application development in general. That means, for applications which have been developed on the old project basis, that they have to be adjusted to this new project structure.

In 6.30, there were five project types: EJB Project, EJB Assembly Project (EJB-JAR), Web Project, Web Application Project (WAR), and Enterprise Application Project (EAR). As of SAP Web Application Server Release 6.40, the J2EE application developer only has to handle three project types for one Web application: An EJB Module Project (EJB Project and EJB Assembly Project are merged), a Web Module Project (Web Project and Web Application Project are merged), and an Enterprise Application Project (no change) which is still the project type for creating a deployable file for the whole application.

The 6.40/NW04 Developer Studio version cannot handle the old project types, which means that all J2EE projects have to be adjusted manually to the new structure. This guide gives a detailed description of these necessary steps.

Migration Process

For the migration process itself, the procedure can be divided into the following main necessary steps:

Extract information from the old projects

1. Import the 6.30 J2EE projects into the SAP NetWeaver Developer Studio if they are not part of your workspace yet. You import projects via File → Import → Multiple Existing Projects into Workspace.

2. In the next wizard screen, enter or select the base folder which contains the projects you want to import. Choose the single projects for your import, and set flag Open Projects after Import. Press Finish to start the import process.
3. Open the J2EE Explorer view via Window → Show View → Other → J2EE.

4. Open the application.xml of your Enterprise Application Project by double-clicking it. This .xml file is part of your EAR project:

Create new projects

1. Open the Navigator view via Window → Show View, and copy the sources of the projects via the corresponding context menu entry. If you want to use the same names for the 6.40 projects just rename the old ones via context menu entry Rename before you start the copy process.

2. Enter a name for the project copy in the upcoming popup:
3. Create a new EJB Module Project via File → New → Project → J2EE with any project name.

4. Also create a new Web Module Project and a Enterprise Application Project via the same menu path.

**Handling of complex project structures**

Concerning the handling of complex project structures you have to consider: In the 6.30 Developer Studio you could define a M:N relation between EJB Projects and EJB Assembly Projects or between Web Projects and Web Application Projects. This is no longer possible. The following description assumes that there is a 1:N relation between those projects which is the case for most users.

In the other case, which means that the same EJB Project/Web Project is referenced in several EJB Assembly Projects/Web Application Projects, there are two possibilities to handle this:

   a. Create one EJB Module Project per EJB Assembly Project (as described above). Later, you would copy the contents of the one EJB Project into every EJB Assembly Projects which results in duplicated code and is not recommended.

   b. Create only one EJB Module Project per all EJB Assembly Projects, contrary to the standard solution. In the IDE, you would then have only one configuration any more. Further configurations can be created by importing the .ear file into the deploy tool.

**Copy the contents of the old projects into the new projects**

Open the Navigator view, and copy the sources of the projects via the corresponding context menu entry (right mouse click). It’s also possible that you execute the copying steps in your file system.

1. **The old EJB Project**

   a. Copy the folder *ejbModule* of your 6.30 project into your new 6.40 EJB Module Project. Overwrite existing files.

   b. Merge the `.classpath` files of the 6.30 EJB Project and the new 6.40 EJB Module Projects. Do the following steps:

      i. Open both `.classpath` files.

      ii. Copy the classpath entries of the old project with the kinds `var` and `lib` which are not already part of the 6.40 project. Take care to avoid duplicates with different names (e.g. the same library is added with full path and with variable).
If in the old .classpath file classpathentry is referring to other EJB Projects, there are two possibilities to refer to the new one(s):

i. Write manually classpathentry to the used EJB Module Projects or

ii. Add a reference from the EJB Module Project to the used EJB Module Project:

- Open the context menu of the new EJB Module Project, and choose Properties.
- Choose Java Build Path and the tab Projects.
- To refer to the EJB Module Project, select it, and press OK.

If in the old 6.30 EJB Project the .tssapinfo file exists, merge .tssapinfo into the EJB Module Project. This means that you copy the tags <AddLibs> of the old file into the new one, the other tags already exist.

2. The old EJB Assembly Project

a. Copy folder META-INF into the new EJB Module Project, and overwrite existing files.

b. Merge the .classpath files of the old 6.30 EJB Assembly Project and the new EJB Module Project as described before. Take care that you do not merge the entries for the referenced EJB Projects which do no longer exist.

3. The old Web Project

a. Copy the folders source and webContent into the new Web Module Project, and overwrite existing files.

b. Merge the .classpath files of the old 6.30 Web Project and the new Web Module Project as described before.

If in the old .classpath file classpathentry is referring to other EJB Projects, there are two possibilities to refer to the new one(s):
i. Write manually \texttt{classpathentry} to the used EJB Module Projects or

ii. Add a reference from the Web Module Project to the used EJB Module Project:

- Open the context menu of the new Web Module Project, and choose \textit{Properties}.
- Choose \textit{Java Build Path} and the tab \textit{Projects}.
- To refer to the EJB Module Project, select it, and press \textit{OK}.

If in the old Web Project the file \texttt{.tssapinfo} exists, merge the files \texttt{.tssapinfo} of the Web Project and the Web Module Project. This means that you copy the tags \texttt{<AddLibs>} of the old file into the new one. The other tags already exist.

4. The old Web Application Project

a. Copy the folder \texttt{WEB-INF} into the new Web Module Project, and overwrite existing files.

b. Merge the \texttt{.classpath} files of the old Web Application Project and the new Web Module Project as described before. Do not merge the entries for the referenced Web Projects which do no longer exist. Replace the entries for the referenced EJB Projects, if they exist, by references to the new EJB Module Projects.

5. The old Enterprise Application Project

a. Copy the folder \texttt{META-INF} into the new 6.40 Enterprise Application Project, and overwrite existing files.

Refresh and rebuild all migrated projects. Next, you have to adjust the references to fix the errors that might still occur.

\textbf{Adjust references to other projects/libraries}

Open the \textit{J2EE Explorer} to complete the new 6.40 Enterprise Application Project.

1. Open the file \texttt{application.xml}, choose the tab \textit{Source}, and delete all referenced modules \texttt{(<module>...</module>)}.

2. Select the Enterprise Application Project, and choose \textbullet Add Modules from the context menu.

3. Select all Web Module Projects and EJB Module Projects which shall be added to this project (corresponds to the old Web Application Projects / EJB Assembly Projects). Press \textit{OK}.

Refresh and rebuild all projects again, now there must not occur any error.
Web Dynpro Project Migration

This chapter describes the procedure for migrating Web Dynpro applications that were developed with the Web Dynpro design time tools as they were shipped with the SAP Web Application Server Release 6.30 and which are currently running on a 6.30 SAP J2EE Engine.

A Web Dynpro migration project consists of a few tasks. The following list gives you an overview of all the steps that are necessary. For detailed instructions about each step, see the corresponding sections.

Purpose

Web Dynpro as it is shipped with the SAP Web Application Server Release 6.40 is a more advanced and versatile technology than the 6.30 version. For example, with the 6.40 version a new RFC type, the so called Adaptive RFC, is shipped, including a new model type which supports the import of functionality from an R/3 backend. Using this new RFC, changes on structures in the backend system do not affect the Java Web Dynpro application, for example. Also new tools are available with the new release, e.g., the Data Modeler which provides graphical support for the definition of the mapping scenarios within the Web Dynpro application.

Migration Process

For the migration of your 6.30 Web Dynpro project files, there are a few steps you have to execute. The Web Dynpro design time tools offer a set of functions for your migration support, keep the sequence of the following single steps:

1. **Import the 6.30 Web Dynpro Project** into the SAP NetWeaver Developer Studio.
   For an import of a single project, goto *File → Import → Existing Project into Workspace*. If you would like to import more than one project in one step, choose *File → Import → Multiple Existing Projects into Workspace*.

2. **Repair the WD Project**:...
By executing the project repair function *Project Structure and Classpath*, missing folders are created, and classpath entries are added.

3. Start the **Activate RAW Language Support** on the imported project:

As a result, the raw language support is added to each user interface entity of the Web Dynpro application, which are the views, as well as to each simple Dictionary type in one step:

4. **Make a Refresh on the Project.**

Switch to the **Navigator** view, and choose context menu entry \(\text{Refresh}\) on the project name.
5. **Rebuild** the project

   From the **Web Dynpro Explorer**, start the Rebuild of the WD project (right mouse click).

In addition to the steps described above, further developer actions might be necessary to finish the migration project. For these final procedures, read the following paragraphs which describe optional actions.

**Developer Studio - Actions**

After the 6.30 project migration to the 6.40 landscape it might happen that you will get import errors which you have to manually resolve using function **Organize imports** from the Implementation tab of each application entity.

The possibly remaining import warnings are due to a known compiler option and can be turned off via **Windows → Preferences → Java → Compiler → Problems → Unused imports: Ignore**.

**UI Elements - Actions**

With Release 6.40, there are first more UI elements available than with 6.30, and second the UI element properties may vary from those of the 6.30 elements. To check the default settings of the elements we recommend that you check the UI element interfaces using the Rational Rose tool.
And following you find more information about possible action items concerning the Web Dynpro user interface elements:

UI elements now have typesafe `bind*()` methods with `IWDNodeInfo` or `IWDAAttributeInfo` as parameters. This makes existent `bind*(null)` calls ambiguous, they must be changed to `bind*((String) null)`. The compiler reports which calls need to be adjusted.

For the UI elements **scroll container**, **group**, and **tray**, you should check the scrolling properties since there is a change of the default value of the attribute `scrollingMode` from `AUTO` to `NONE`. If you want to have an automatic scrolling behavior of these containers, you have to change the `scrollingMode` back to `AUTO`.

If you are using the scrolling mode **BOTH** or **NONE**, you won’t have to do anything.

**Simple Types - Actions**

Access to the value sets of `ModifiableSimpleTypes` has changed. In 6.30, the sequence `modifiableType.getSVServices().getValues()` always returned a modifiable value set. This had two disadvantages: A cast was needed because the method returns an `IsimpleValueSet`, and there was no `ModifiableType` without a value set. Now, this method will return the original unmodifiable value set or even `null`, if the original type has no value set, and you have to use `modifiableType.getSVServices().getModifiableValueSet()` to get a modifiable variant. See the Javadoc for more information.

**Deployment**

The access to the deployable object part has been moved from `IWDComponentInfo` to `IWDComponent`. 
**Configuration Aspects**

**Developer Studio Configuration**
As of 6.40, the installation of the SAP Netweaver Developer Studio disables the Update Manager feature in Eclipse.

Furthermore, the Concurrent Versions System (CVS) plugin is no longer part of the IDE.

**Web Dynpro Configuration**
In case you access WD Configuration in the framework, your paths might have changed. Instead of method `getConfiguration("sap.com/WebDynpro")`, you might have to use `getConfiguration("sap.com/tc~wd~dispwda")`.

**Portal integration**
As of 6.40, there is an extended API for `WDPortalNavigation.navigateAbsolute()`. The new parameters could be `null`.

**SLD Integration**
If you are using the Web Dynpro Explorer for maintaining your SLD references, you need to call it with a different URL. The new URL is: `http://<host>:<port>/webdynpro/dispatcher/sap.com/tc~wd~tools/Explorer`. The tool should be available with your initial installation, it now runs with user authentication.

The SLD properties have to be maintained in the tool Visual Administrator. You make the corresponding entries within the Visual Administrator at SLD Data Supplier → CIM Client Generation Settings.

[Back to top]
General Restrictions

- The IWDMessageManager Interface has been adjusted, one method was removed. IWDMessageManager.setReportSingle(true) cannot be called any longer.
- All text accessing methods have been moved to IWDComponent.getTextAccessor(). The old methods have been deprecated.

Troubleshooting

Problem:
SAP tools are not accessible within SAP NW Developer Studio

It might happen that the views which contain the SAP tools are not visible within the 6.40 version of the Developer Studio when you are working with a 6.30 workspace. This problem might occur if the target machine for the migration was not the source machine (otherwise, you already would have deleted the complete .metadata directory is described above).

Solution:
Remove the .config folder in your workspace.

Problem:
In the 6.40 environment, a 6.30 WD application which uses a business graphic from the Internet Graphic Server (IGS) is not automatically updated after the context has changed.

Solution:
Implement a method call of graphic.forceUpdate() within the wdDoModifyView method to guarantee that the new business graphic is created.

Further Information

Documentation

For 6.40 J2EE documentation for the development of Enterprise JavaBeans, read the corresponding rampup documentation chapter. Follow the documentation path as following:


You can access the 6.40 rampup information concerning Web Dynpro Runtime and the Web Dynpro design time tools via

Migration Problems
If you need support from the J2EE developer team during adjustment of your J2EE project structures, create a CSN message in component BC-DWB-JAV.
For questions concerning the Web Dynpro project migration, choose CSN component BC-DWB-WD-JAV (IDE problems), or BC-WD-JAV (runtime problems).