THE FAST WAY TO COMPONENT-BASED DEVELOPMENT WITH THE SAP NETWEAVER CE 7.1 – PART III
ORGANIZING TEAM DEVELOPMENT

Applies to:

SAP NetWeaver Composition Environment 7.1.

Summary

SAP NetWeaver Composition Environment 7.1 (CE), SAP is JEE 5.0 certified. SAP has added several enhancements of concepts and object types to the Java development. These can be used optionally. The enhancements address certain development needs, such as a client-independent UI, the handling of database tables, or the creation of composite applications. Conceptual enhancements ensure an improvement in the structuring of applications and the development process.

The development according to the component model is organized by development configurations, which define the development landscape for a specific software development project. Development configurations can be created locally (instead of using the SAP NetWeaver Development Infrastructure).

This third part of the article focuses on organizing team work when using local development configurations. These processes are described in a "hello world" example that was started in part II. All development steps of the component-based development with optional infrastructure require only an SAP NetWeaver Developer Studio and a runtime system. We will now see how to additionally use a file share. Therefore our steps will be to

- Export and distribute a local development configuration,
- Import a local development configuration from a colleague,
- Reuse a software component archive
- Sync the team's work in the file system
- Briefly show how to use a third-party tool for file versioning
- Use SAP's command line tools for building and packaging an application centrally

Author: **Wolf Hengevoss**

Created on: 21 January 2008

Author Bio

Wolf Hengevoss graduated in natural sciences at the University of Kaiserslautern. In 1999, he joined SAP as a member of the product management. He has worked in the Basis group focusing 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 is – apart from SUN, of course – the first certified vendor of a *JEE 5.0* server implementation. Additionally to pure standard J2EE/JEE applications SAP provides you with additional types of Java applications and enhanced control of the development process if you use SAP's component model. In the first part of this article I described the *development scenarios in the SAP NetWeaver Compositions Environment 7.1*, in part II I showed how to perform component-based development as a single user. Now, I want to describe how you can sync the work in a team and enhance the scenario with an optional development infrastructure. First, you can use a file share to have access to all development objects of the team, then, because a local build would require the download of the complete share folder to you local PC you can integrate the optional steps in the command line tool for a central build. In a second step you can integrate your source code versioning.

Prerequisites

For the prerequisites, please refer to part II of this article. Note that this part of the article is based on the Developer Studio of SAP NetWeaver CE 7.1 SP03. Some aspects of the UI will probably change in the future.

Component-Based Development with Optional Development Infrastructure – Integrating a Team

You have set up a local development configuration which is the fastest way to component-based development – all you need apart from the runtime is provided with the Developer Studio, so this helps you integrate into the SAP NetWeaver CE 7.1 environment with least effort. If your team grows you can use:

- A *command-line tool for build and packaging* as part of the Developer Studio installation. ANT tasks for build and packaging are delivered the same way.
- A *source repository* integration (team development support): You can integrate the work of developers in a team using the file system. I will deal with that question in the next part of the article in detail. Direct integration of source code versioning into the development processes of the SAP NetWeaver Developer Studio is possible with SAP's Design Time Repository, which is delivered with usage type DI of SAP NetWeaver 7.0, which fully supports development with the SAP NetWeaver CE – please compare part one of this article.

If you're working in a team, your work has to be synchronized. As stated before, that process is in your hands. Keep in mind that DCs work similar to other projects but are emphasizing the reuse idea. The best way perhaps is to organize work along DC boundaries and find an agreement on the DC interfaces, the pubic parts.

Exporting, Distributing, and Importing a Development Configuration

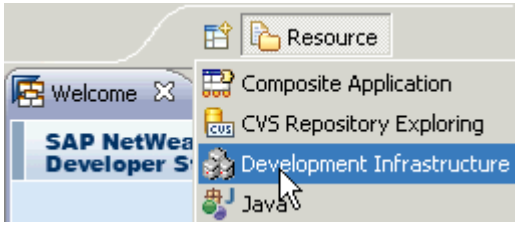
Exporting a Development Configuration File

First thing to synchronize work in a team is to start with the same state. This is easy: The team lead – or the first one who started development – can simply export the development configuration and put it on a file share or an email for his or her colleagues, who import the file. We'll use the already existing development configuration J2EE_Standard_Dev (if you have started with this part III, please refer to part II).

1. In the Developer Studio navigate to the *Development Infrastructure* perspective to export your development configuration (here you just opened the Developer Studio with a new Eclipse workspace).

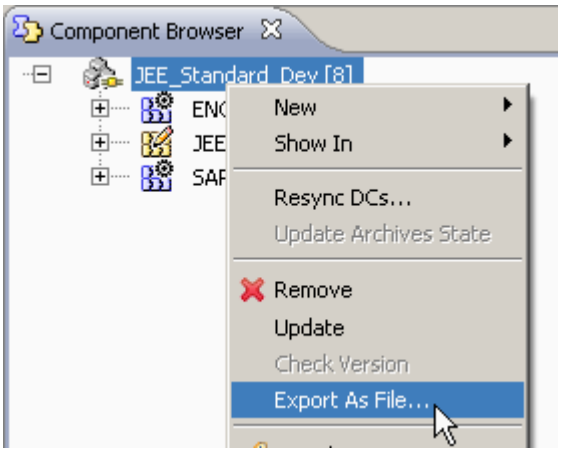
   a. Choose ***Open Perspective***:

   b. Choose the perspective ***Development Infrastructure***.

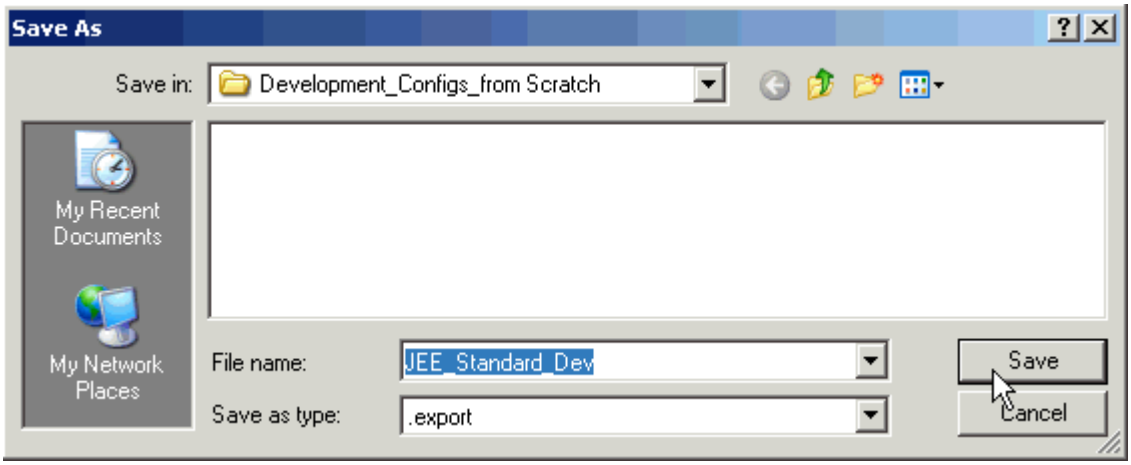


   You see the *Component Browser* view (if not, in the menu choose *Window* → *Reset Perspective*).

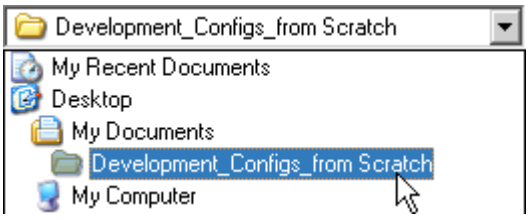
2. In the context menu of your *development configuration* choose ***Export as File…***



3. In the following dialog screen accept the name for the export file with ***Save***:



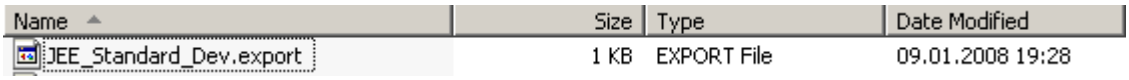
The development configuration export file will be written to a folder ***Development_Configs_from Scratch*** under your *My Documents* folder:



Distributing a Development Configuration File

Now, you can make the file available to your colleagues.

1. Navigate to the folder for development configurations in the file explorer:

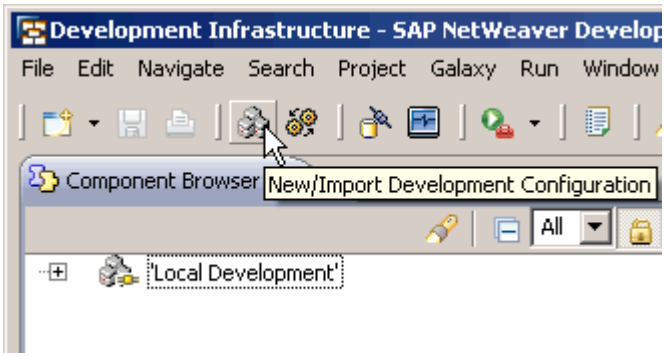


2. From here, you can handle the export file of your development configuration like any other file, put it on a share drive or send it to your team via email.

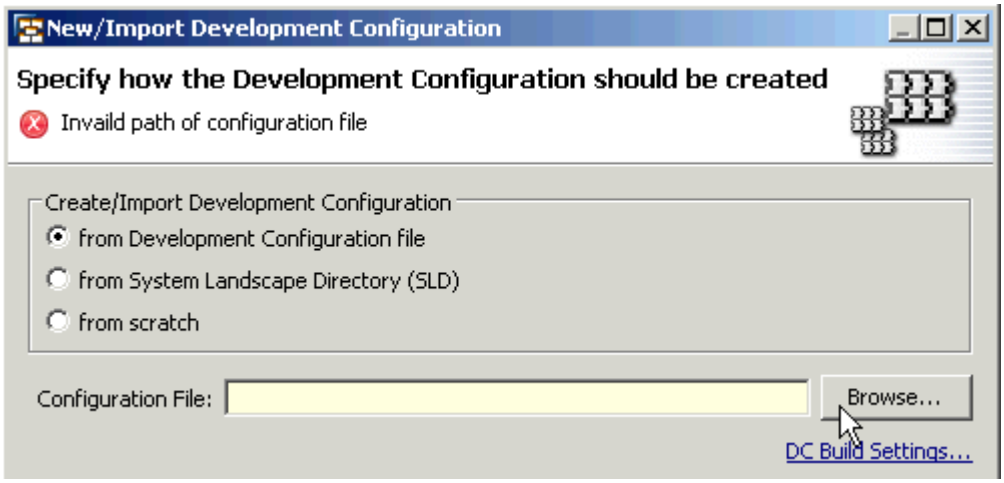
Importing a Development Configuration File

To work with the same starting conditions as a colleague, you import the same development configuration.

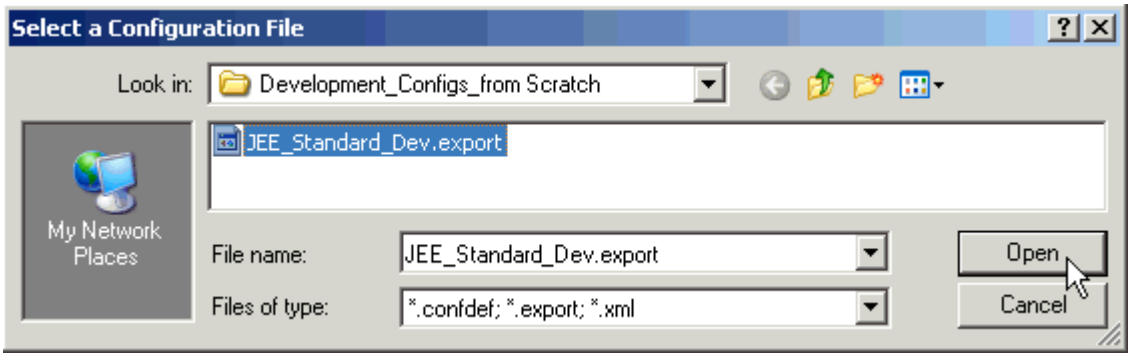
1. In the Developer Studio navigate to the *Development Infrastructure* perspective under *Window* → *Open Perspective* → *Other* → *Development Infrastructure*.
2. To import a *development configuration file* provided by a colleague there are two ways:
   a. If you got the file for example via email, copy the file to the folder ***..\My Documents\Development_Configs_from Scratch*** (where any development configuration you create is stored and looked for by default); in principle any other folder is also possible.
   b. If the file is available on a common *share folder*, get the path.
3. In the *Development Infrastructure* perspective choose *New/Import Development Configuration*:



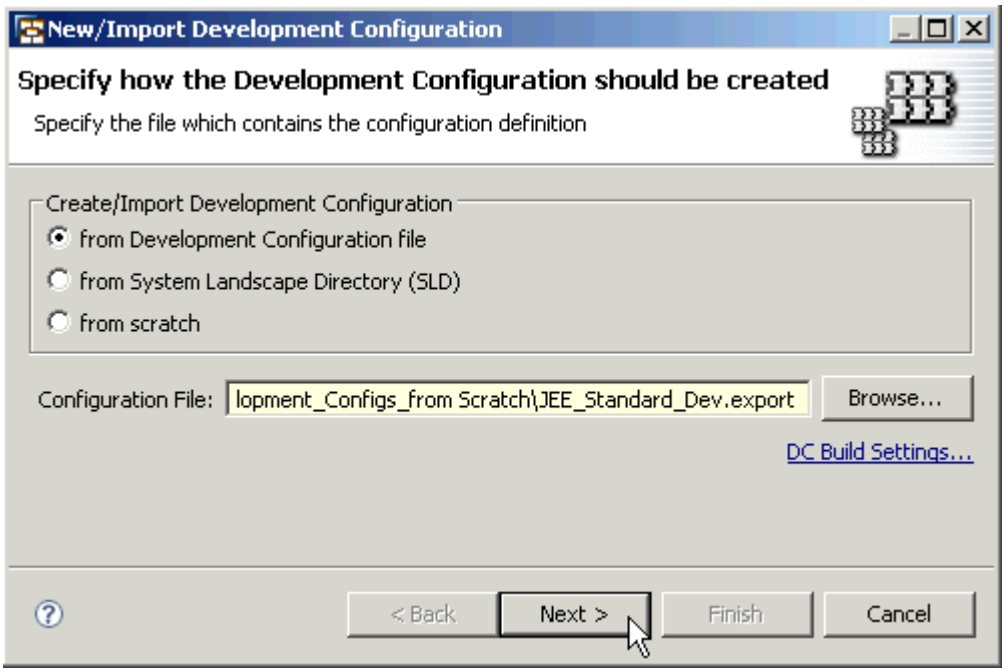
4. In the next dialog, choose to import *from Development Configuration File*, then choose ***Browse***:



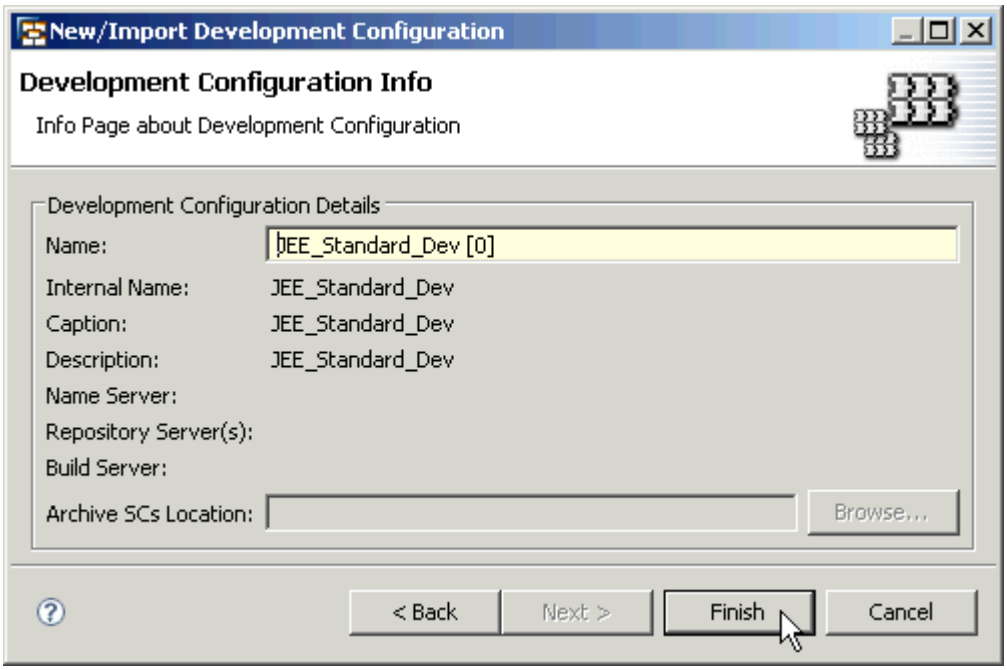
5. In the next dialog window select a development configuration (by default the folder ***..\<your user>\My Documents\Development_Configs_from Scratch*** is opened):



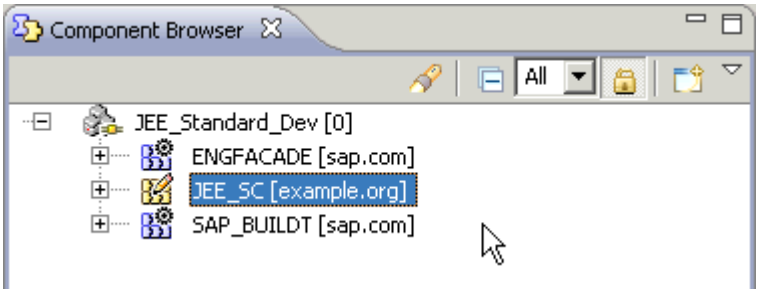
6. You are back on the initial dialog window. Choose ***Next***.



7. On the *Development Configuration Info* dialog to confirm choose ***Finish***.



8. In the *Development* Infrastructure perspective open the *Component Browser* view. The imported development configuration is available now:



When you now close the Developer Studio the settings including the development configuration you loaded will be stored.

NOTE

This does not mean, that you already see the work of your colleague but only that you work with the same settings – we will take care of the in the next step.

Btw: Be careful to use the same version of the SAP NetWeaver Developer Studio as your colleague! You can check the version if you choose *Help* → *About…* in the Developer Studio's menu. The version of the Developer Studio *and* the used SCs must match with the target runtime system. This is especially important when developing with Web Dynpro Java.

Using an SC Developed in Your Team

To be able to make use of an SC that was not delivered with the Developer Studio you have to create a development configuration or load one of created by a colleague and add its content to your system.

NOTE

This part is meant for development where software is handed over from one developer to the next. The file system is the minimum level of integration we see as a prerequisite for team development. Optionally, you can put the content under version control, using e.g. SAP's *Design Time Repository* or open source *CVS*. This will be explained in the next section.

We have to differentiate between two possibilities here:

- You can build on top of the existing software components not changing the content of these SCs
- You want to change existing development components of the SCs

Building on Top of Existing Software Components

There are two ways to enhance existing functions:

- Use the SCA delivered and add more DCs directly (you even can delete and replace DCs) or extend existing DCs (in the latter case the DCs to be extended must contain their sources)
- In the development configuration just create a new SC using the one delivered. This will of course mean to change the development configuration.

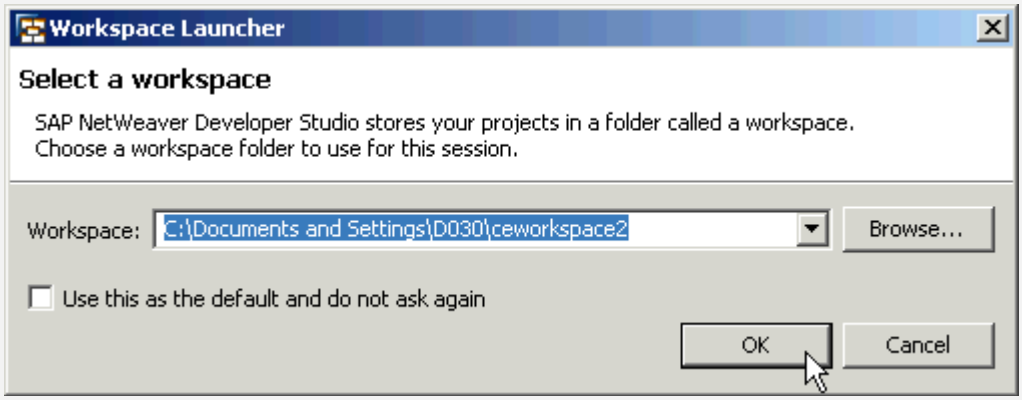
In each case you create a new SC archive in the end.

NOTE

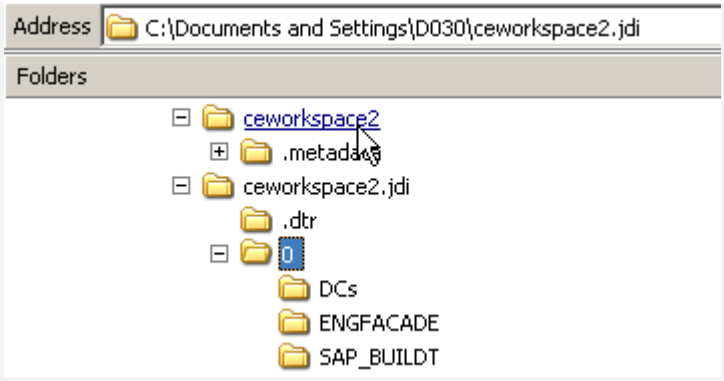
One reason to use an additional SC is for example that you then may decide to deliver on SCs including the source code, the other just as archives. Another possible reason is that in this case you leave the original SC untouched.

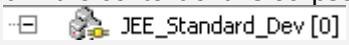
NOTE

When you opened the Developer Studio you needed to specify an *Eclipse workspace*, e.g.:



This workspace is represented in the file system. If you navigate there, you see the folder of the chosen *workspace* containing the Eclipse *metadata* and one folder called *<workspace name>.JDI*. Here, you see the numbered list of loaded development configurations containing folders – or *compartments* – for all software components (SCs) of your development configuration. For the SC currently under you see the folder ***DCs***:

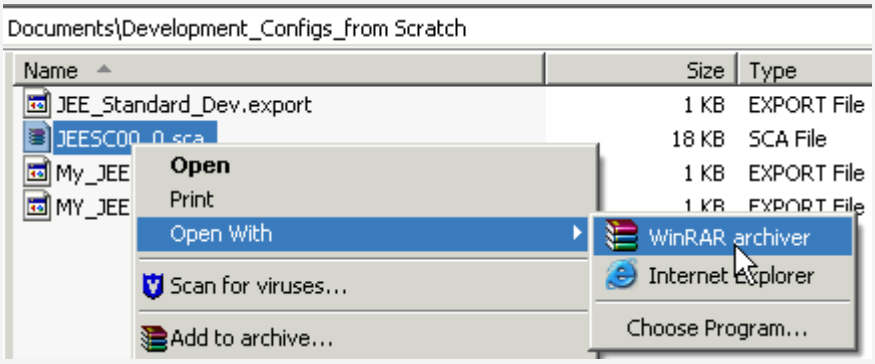


The number indicates the number of the development configuration loaded in the context of this eclipse workspace. The same number is also shown in the *Component Browser*: 

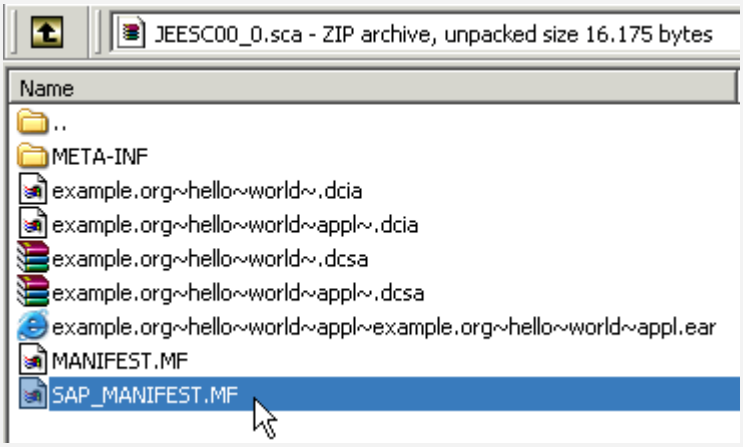
1. Copy the SCA file of the SC you want to develop into a folder for example *<drive>:\Documents and Settings\<user>\My Documents**Development_Configs_from Scratch***.

NOTE

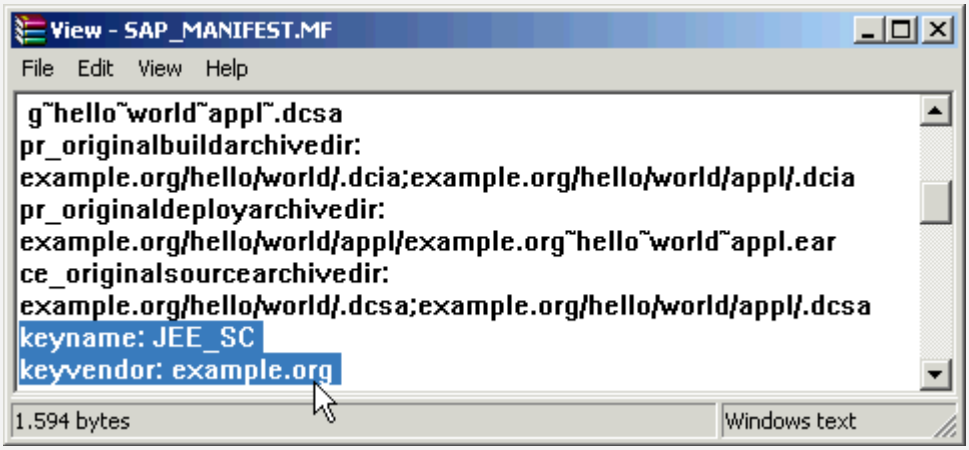
1. To see details of an SCA file open it with *WinZIP* or *WinRAR*:



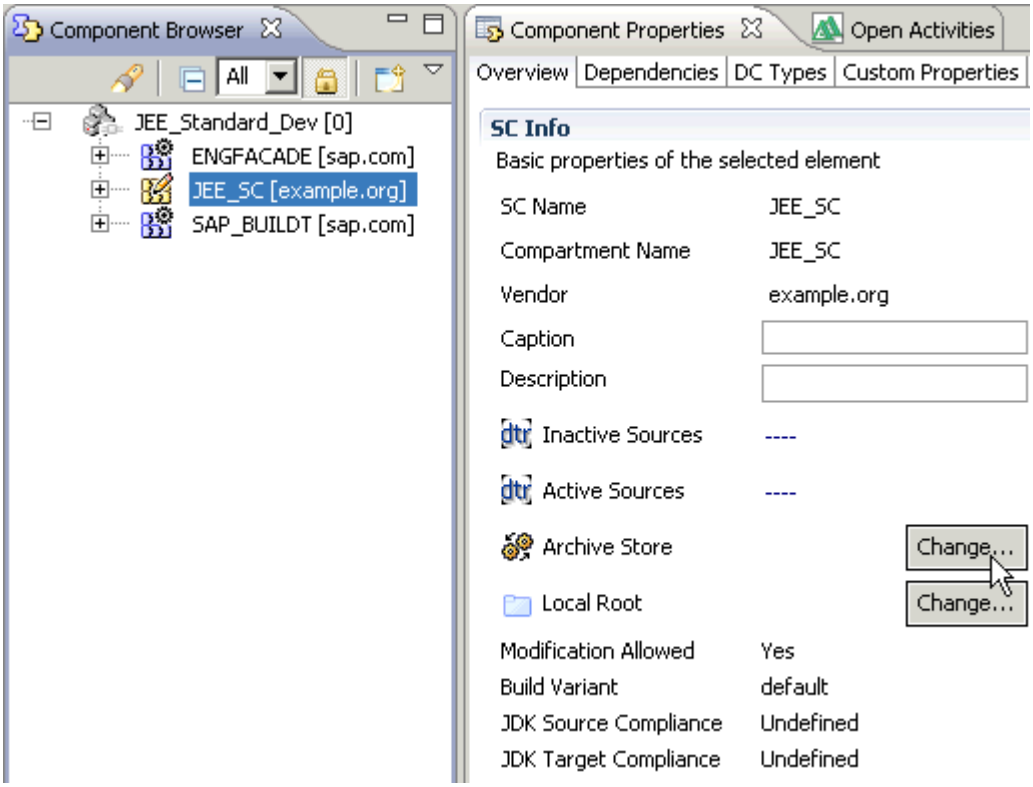
2. Open the file ***SAP_MANIFEST.MF*** by double clicking:



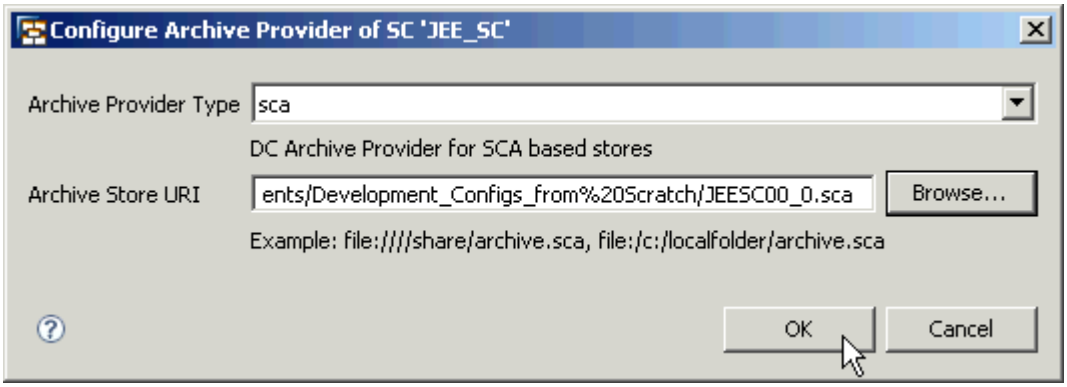
3. Read the ***SC name*** and ***vendor*** information if needed:



2. In the *Component Browser* select the SC for development and open the tab ***Component Properties*** to define the ***Archive Store*** settings:



3. Choose to *Change* the *Archive Provider Type* to ***sca,*** *browse* to the folder where the SCA file is stored and confirm with ***OK***.

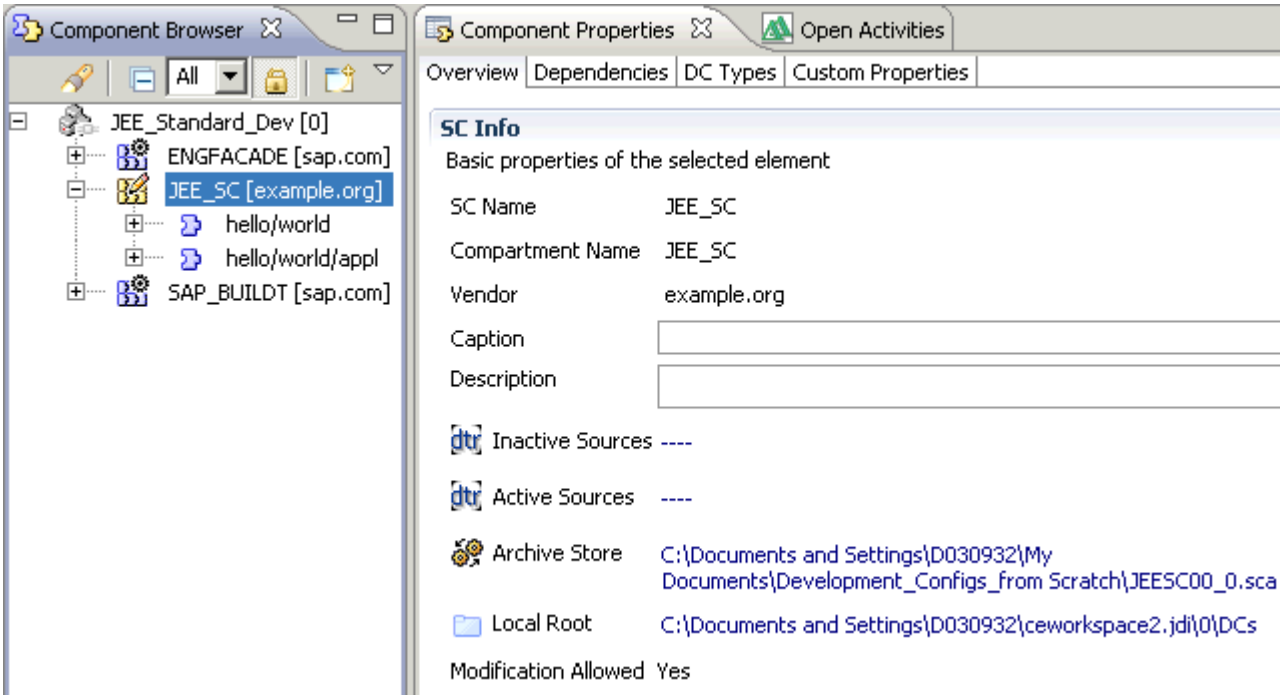


NOTE

Other settings for the *Archive Provider Type* are

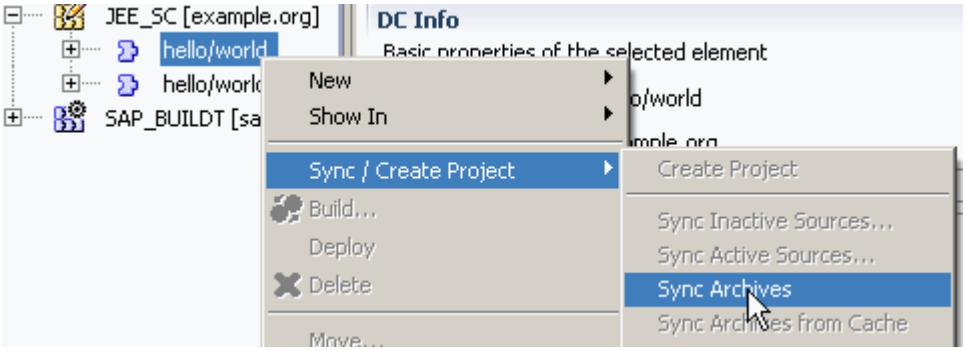
- ***arcpool*** (used to load DC archives from the archive pool delivered with the Developer Studio)
- ***fs*** (used to load DC archives from the file system to reuse existing build results, e.g. \\share\DCs\a.dcia etc.)
- ***local only*** (just tells that all DC archives used are avaialble in the compartment locally)

4. Open the development configuration previously imported in the *Component Bowser* again:



You see that SC *JEE_SC* contains two DCs now. You can work with these DCs not quite the same way as with the ones you created locally: In the next step you see that you can load the archives to use the DCs, but can not create a project (the option is shown to be inactive).

5. In the context menu of the SC you want to develop, you can now choose to ***Sync Archives***:

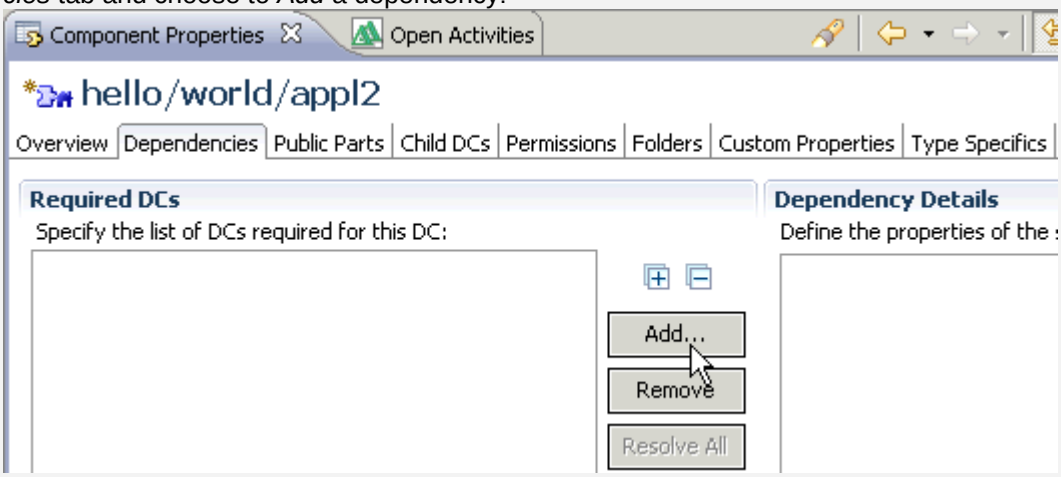


Based on the DCs synced (loaded) you can create new DCs. For example, you can create a new Enterprise Application to use the Web Module DC *hello/world*.

NOTE

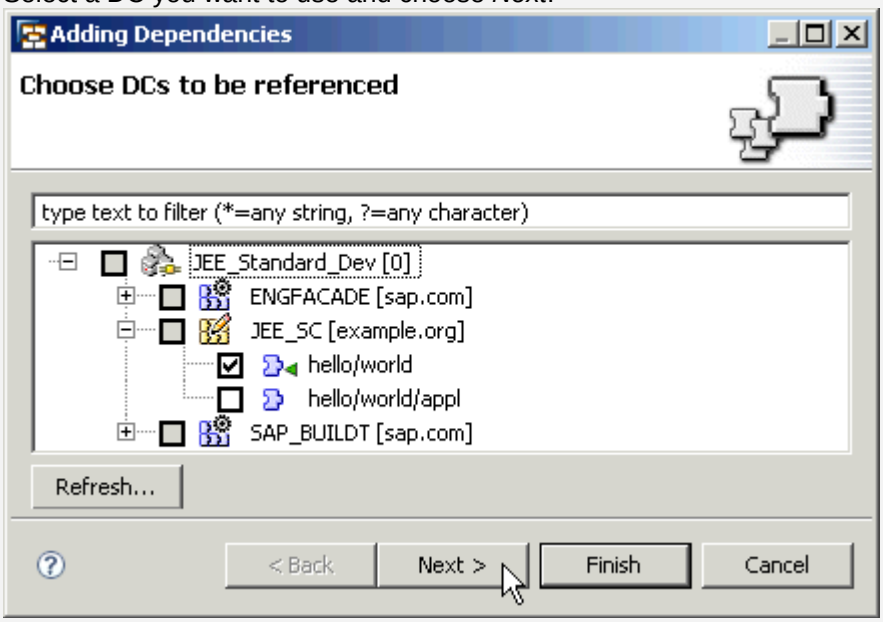
Since you do not have a project available you need to add the Web Module dependency manually to the Enterprise Application. According to SAP's component model this is done by directly declaring a dependency to the public part of the DC you need to use (also compare part II of this article series):

1. After creating an Enterprise Application DC, choose not to switch to the respective JEE perspective.
2. In the *Component Properties* view for the new Enterprise Application choose the *Dependencies* tab and choose to *Add* a dependency:

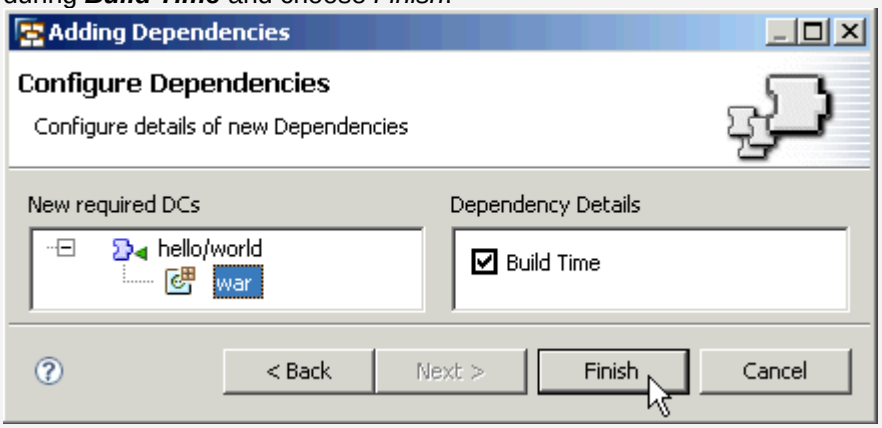


This function might not work in the SAP NetWeaver Developer Studio of the SAP NetWeaver 7.1 Composition Environment SP3. In that case, a Studio update will help. How to deal with development configurations if you update the Developer Studio is described in part II of this article.

3. Select a DC you want to use and choose *Next*:



4. To define the *Dependency Details* choose to use the public part "***war***" of purpose *assembly* during ***Build Time*** and choose *Finish*.



When you build it the Enterprise Application DC will now wrap and deploy the content of the Web Module DC.

The new DCs you created can be exported in a new version of the SCA file.

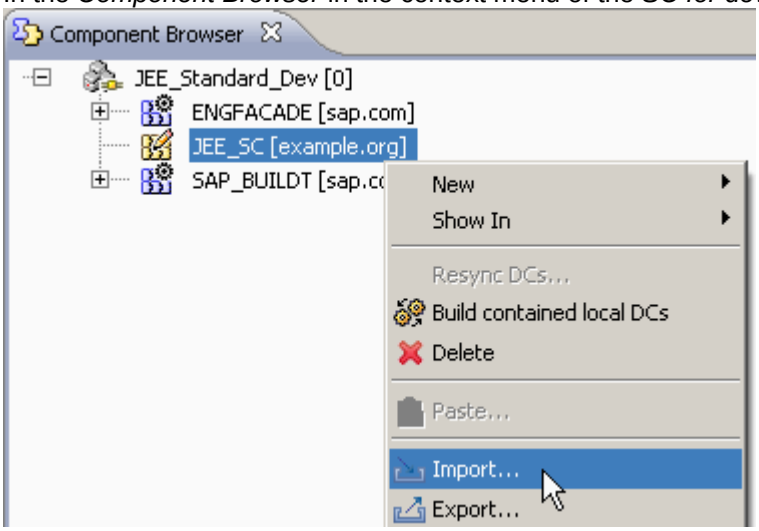
Changing Development Components

Changing existing DCs means to change the containing SC also.

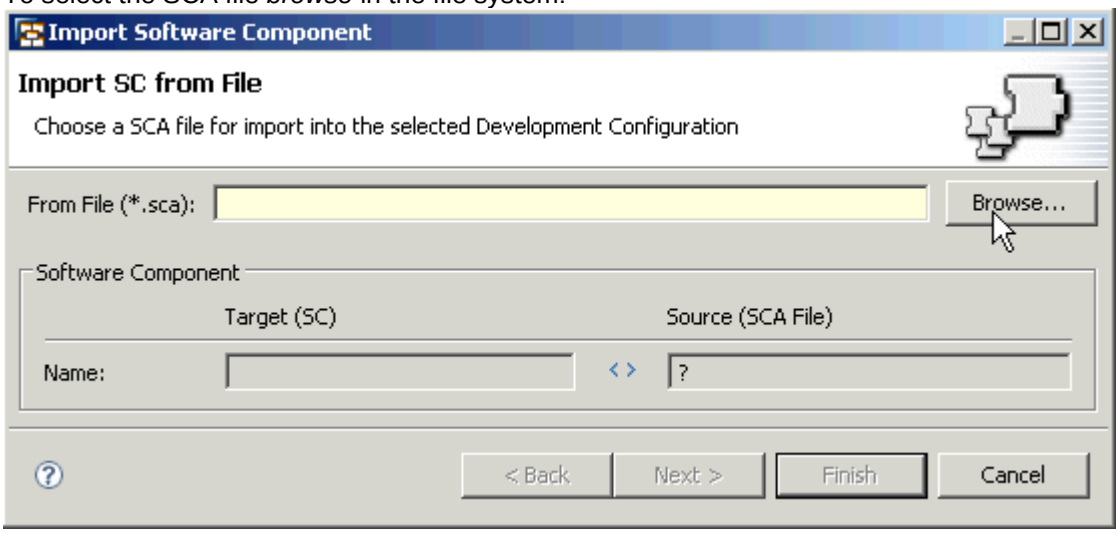
NOTE

The SC needs to have been exported including the sources to make this scenario work. Please refer to part II of this article series for details.

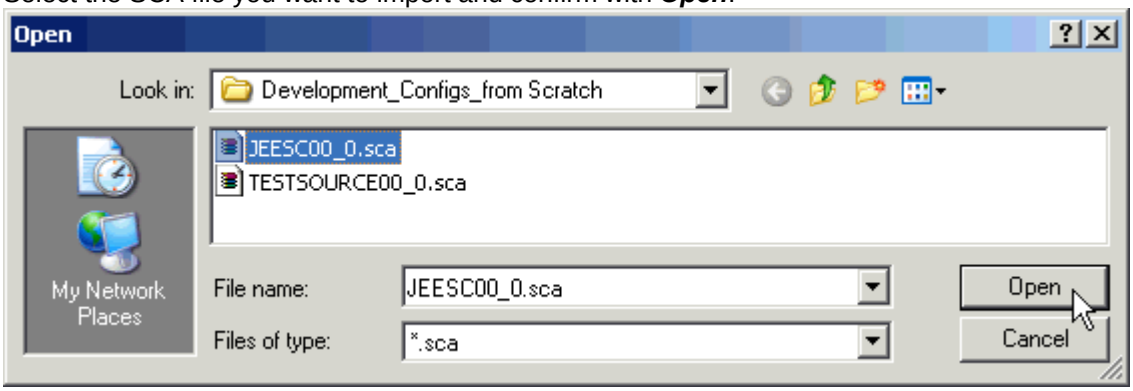
1. In the *Development Infrastructure* perspective import the development configuration as described above.
2. In the *Component Browser* in the context menu of the *SC for development* choose ***Import***:



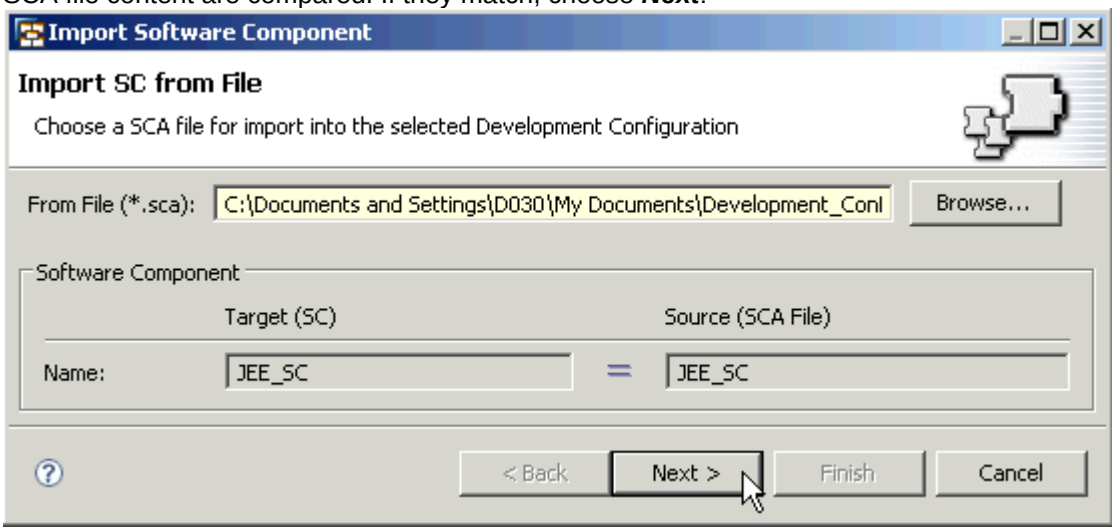
3. To select the SCA file *browse* in the file system:



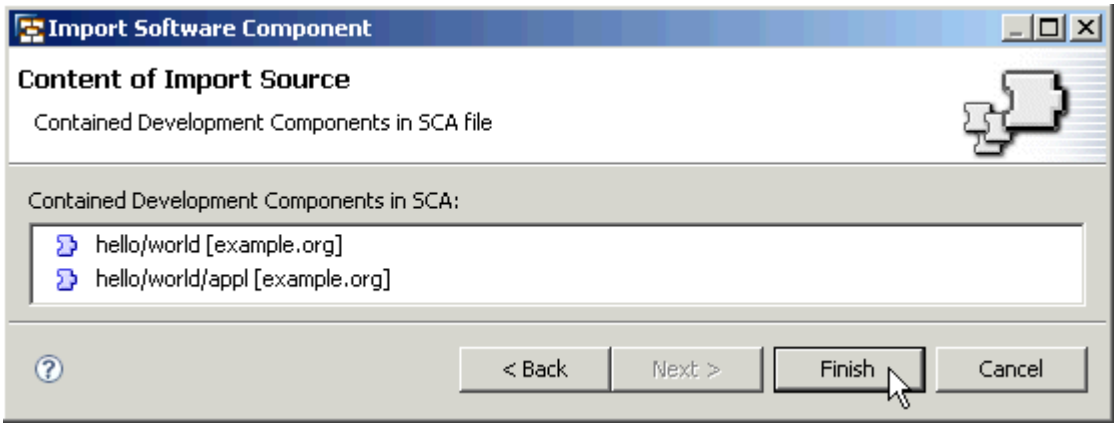
4. Select the SCA file you want to import and confirm with ***Open***:



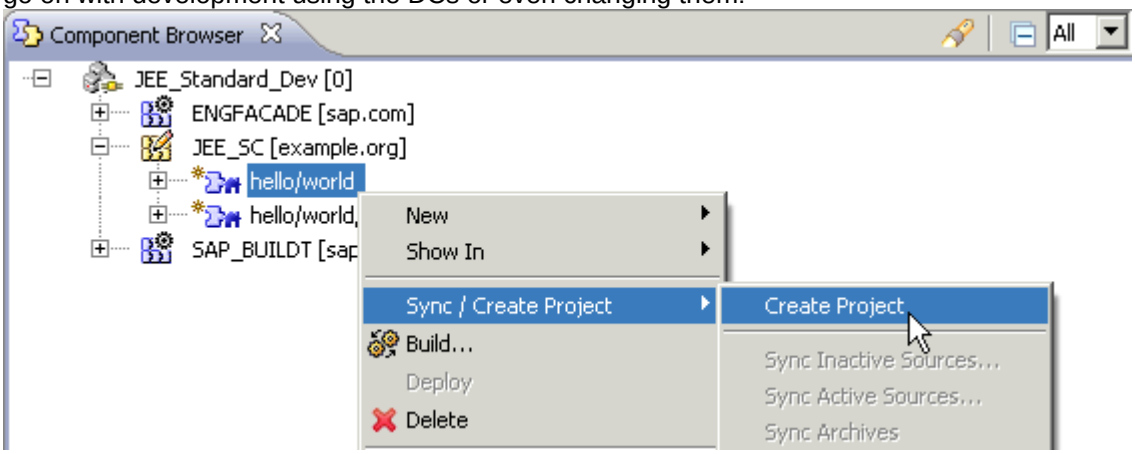
5. Target SC – the name of the SC compartment into which you choose to import the SCA file – and SCA file content are compared. If they match, choose ***Next***:



6. You see the DCs contained in the SCA file. Choose ***Finish***:



7. In the Component Browser you can now not only sync the DC's archives but can create projects and go on with development using the DCs or even changing them:



You can export the changed software component in a new version of the SCA file.

As you have seen, different developers can contribute to the same SC, but only in a sequential manner. To overcome this, we need to make use of a file share.

Working on a Software Component in a Team

So far I have described how to develop software alone – even if it is easy to distribute it in the form of SCAs others can use as a basis for their work or even change to create a new version of it. However, in many cases it is desirable to develop in a team.

There are some points you need to be aware of:

- This means that you need to work on the same set of files or even the same individual files concurrently. Therefore, you definitively need a common place to store the source files.

  NOTE
  In the simplest case this would be a file share to store objects only once in a team and do not work on duplicates as described so far. Due to the problem of overwriting one another's changes and certain limitations in the use of a simple file share I highly recommend the use of a version control system to take care of versions.

- A build process is needed, which gathers the changes of all team members in a final version of the software component.

For the management of the application as a whole, SAP provides a command line tool to deal with all the tasks needed to manage the teams work on the level of build, assembly, etc.

Assigning Development to a Versioning System

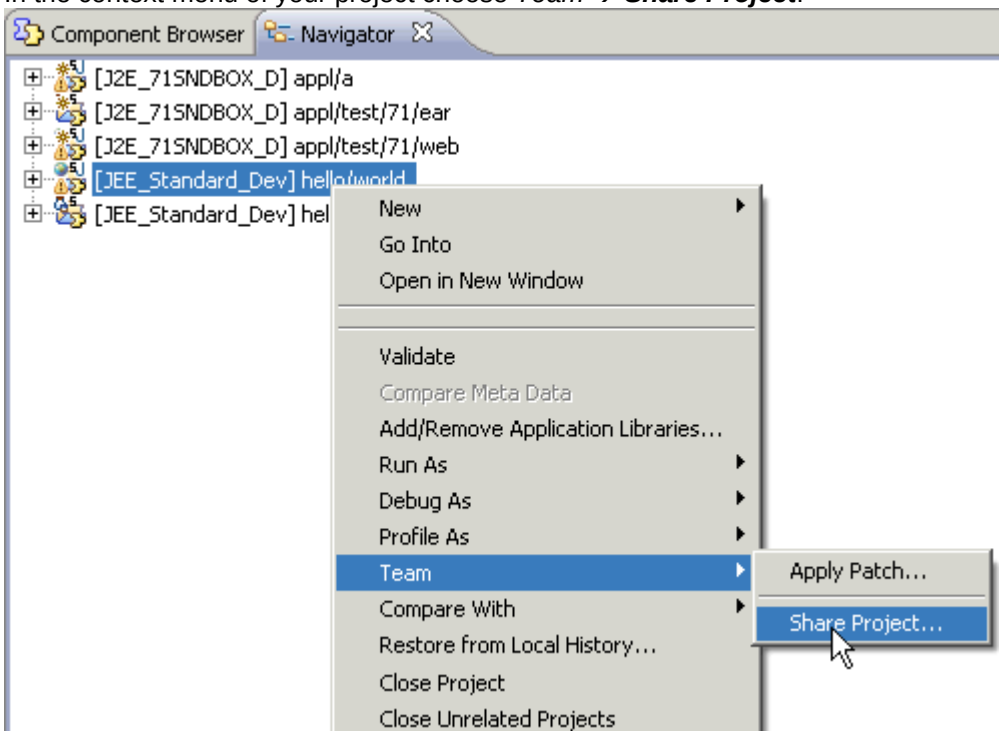
To have versioning of your sources enabled, optionally, you can assign development to a versioning system. This can be SAP's Design Time Repository or CVS.

Prerequisite is – obviously – that the versioning system has been installed and you have access to the system. I will use CVS as an open source, non-SAP tool to show this type of integration.

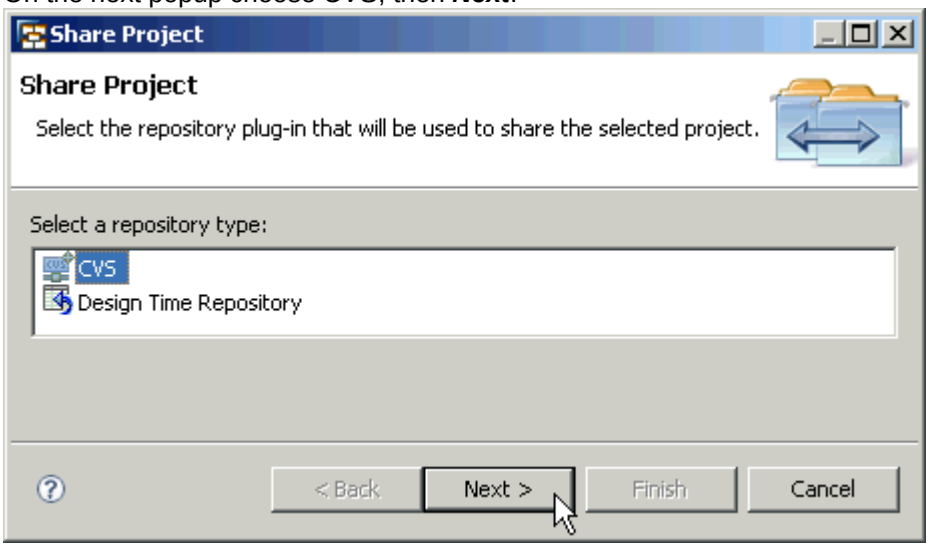
NOTE
This is documented in the Developer Studio documentation in the *SAP NetWeaver CE Developer Studio Documentation (SAP NetWeaver CE 7.1 SPS3)* under *Developer's Guide → Using the Development and Production Infrastructure → Working with the Development Infrastructure → Concepts → Development Scenarios with Development Infrastructure → Using Third Party Development Infrastructure*:

You have created projects already that you want to put under version control now.

1. In the Development Infrastructure perspective open the *Navigator* view under *Window → Show View → Other… → General →* ***Navigator***.
2. In the context menu of your project choose *Team →* ***Share Project***:



3. On the next popup choose *CVS*, then ***Next***:



NOTE
The Design Time Repository (DTR) is SAP's approach to version control. Based on open standards WebDAV and DeltaV its most important feature is to keep versioning information even when sources are transported into another DTR. The DTR therefore allows SAP to ship applications including the sources and offer a modification concept for customers and partners. To find out more about DTR visit SDN, Quick Link ***nw-di***.

4. Then perform the remaining steps of choosing a repository, defining the project's name, etc.

Your project is now under CVS version control. In the same perspective you can synchronize your project with content from the CVS server, merge versions and so on.

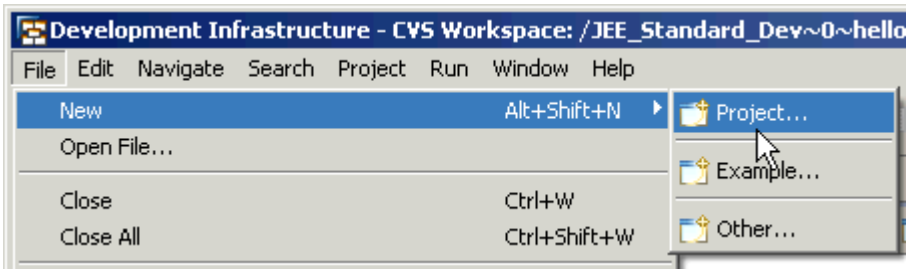
NOTE

You'll find the details on CVS in the Developer Studio under *Help* → *Help Contents* → *Workbench User Guide* → *Tasks* → *Working in the team environment with CVS*.

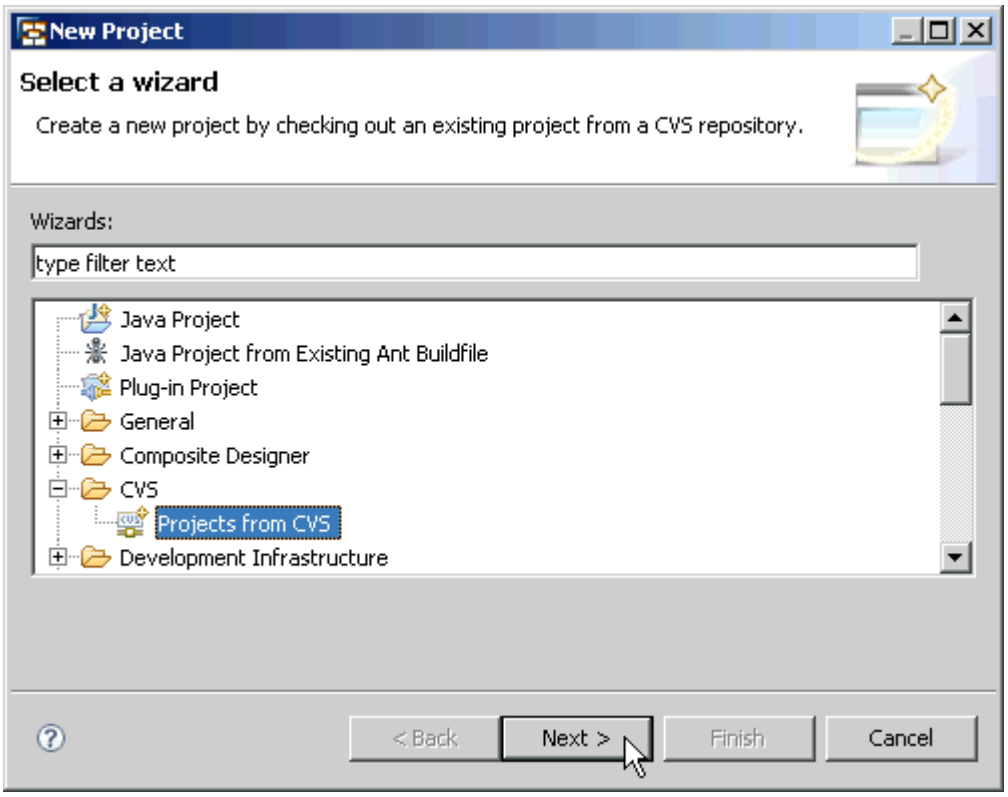
Loading Content from the Versioning System

What has been added to the versioning system can of course be used by other people.

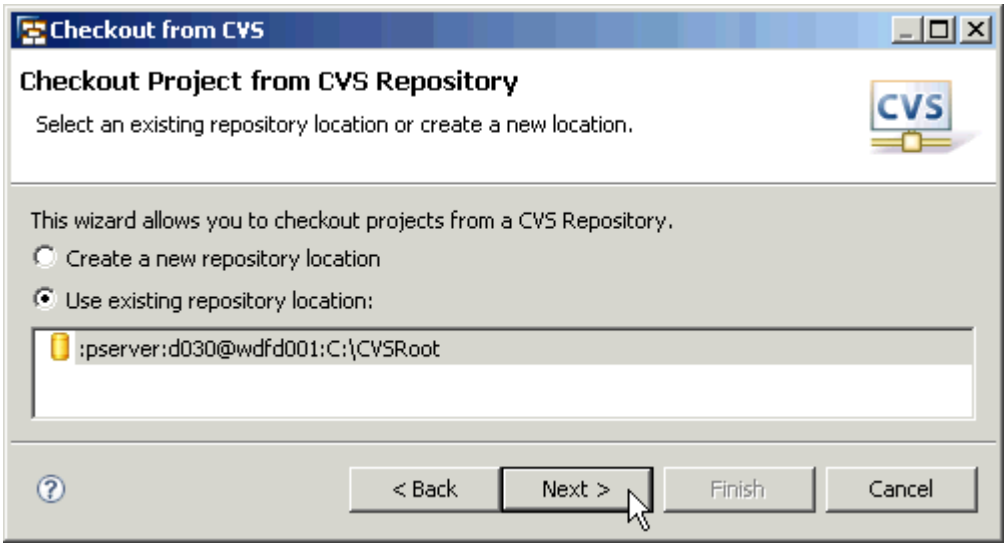
1. In the menu of the Developer Studio choose *File* → *New* → ***Project***:



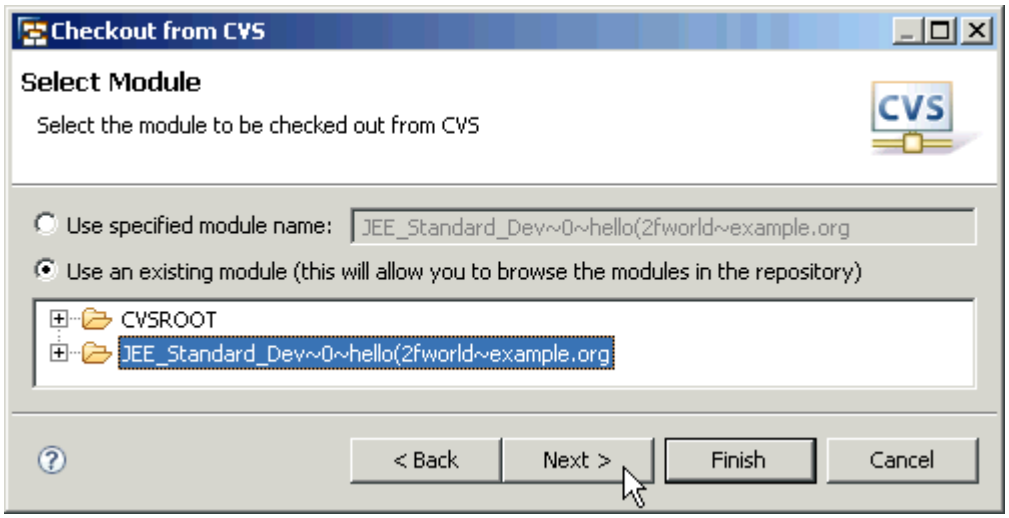
2. Choose *CVS* → *Projects from CVS* and confirm with ***Next***:



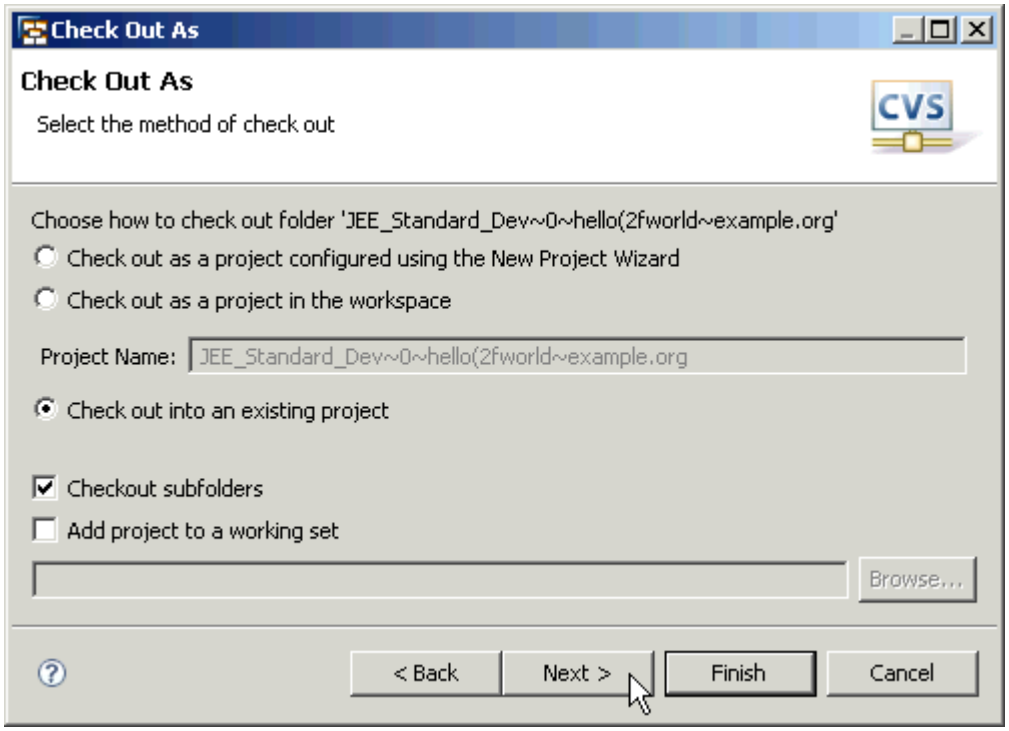
3. Choose a ***CVS server*** and confirm with *Next*:



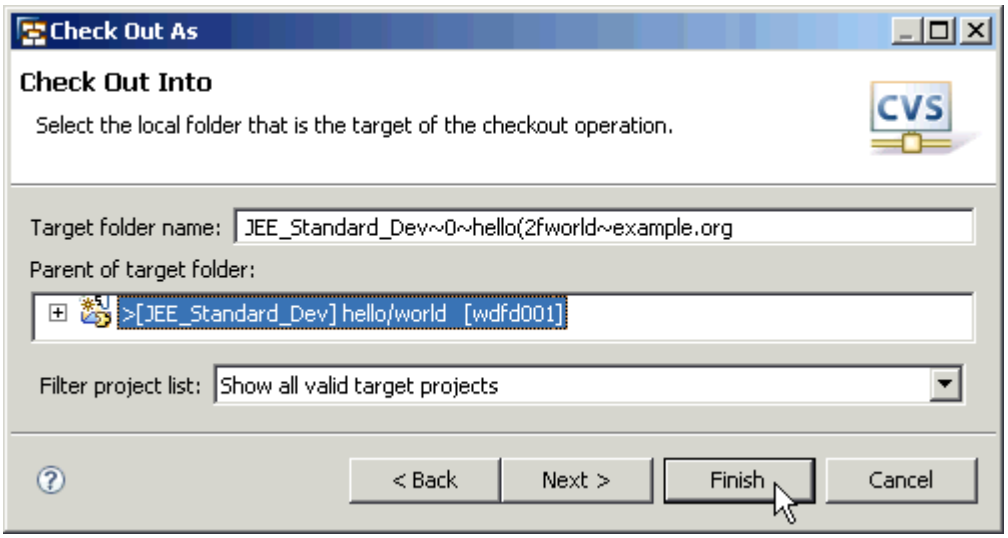
4. Choose to ***use an existing module*** and confirm with *Next*:



5. Choose to ***check out into an existing project*** and confirm with *Next*:



6. Choose the *DC project* you want sync with the CVS and choose ***Finish***.



You can now exchange source states with your team members.

Using a Central Build Process to Gather All Changes Done by the Team

The last step to team development is to take care of the build process. There are in principle two ways to organize this:

- In small projects decide who shall do the build. He or she simply opens the Developer Studio, syncs all DCs and builds them as described in part II of this article series.
- In slightly bigger teams it might be simpler to use a command line tool to build all DCs.

NOTE

Depending on the version of the SAP NetWeaver Developer Studio the command line tool has changed. In older versions you will find the tool in a *tools* folder beneath the eclipse folder of your Developer Studio's installation. Newer versions let you start the tools as described here.

1. Start the command line tool and get familiar with the online help.
2. Use the *buildalldcs* command from the DC command line tool to build all DCs as described in for the newer version.

There are also commands available to package the build results in an SCA file.

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 are already using one – the scenario described here is what you were 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 effort are humble. If you're interested in details, go on reading part II of this article describing the steps of the development process in detail. All objects created in part II of this series of articles can be shared with colleagues – from the development configuration to the software components. If it comes to team development file versioning and setting up central processes becomes important. You can in this scenario even add this type of functions from third-party providers. Of course, it would also be an option then to have a look at the SAP NetWeaver Development Infrastructure for Java-based development in SDN to support the complete life-cycle of development.

Related Content

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

- Previous parts of this series of articles: Part I Development Scenarios; Part II Local Development Configurations.
- SAP NetWeaver Development Infrastructure, which brings in-depth discussions of the NWDI Services and the component model and also contains a video demo of NWDI.
- 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. It is also more and more integrated with Java-based development.

Copyright