

SAP NetWeaver 7.1 Enhancement
Package 1



Tutorial: How to Perform Team-Oriented Development

Document Version 1.00 - August 2009



SAP AG
Neurottstraße 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

© Copyright 2005 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.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Typographic Conventions

Type Style	Represents
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Contents

1	PREPARING THE SYSTEM	2
1.1	Granting Initial Privileges	4
1.2	Creating Users and Assigning Groups Using UME	6
2	CREATING NEW PROJECTS	7
3	ENHANCING THE PROJECTS	10
4	STARTING THE APPLICATION	12

How to Perform Team-Oriented Development

If you work in a team with centrally organized development using the version control provided with the Design Time Repository (DTR), you actually perform team-oriented development. With this tutorial we represent the development process of a small Java EE application in the SAP NetWeaver Developer Studio with the DTR as a source code control system.



Note

This tutorial does **not** describe the steps necessary to implement a central build environment, or how perform the software logistics processes.

By the end of this tutorial you will be able to:

- Set up the Developer Studio to work with the DTR as a version and source control system for each developer in the team.
- Set up the DTR for development and how to work with DTR.

Prerequisites

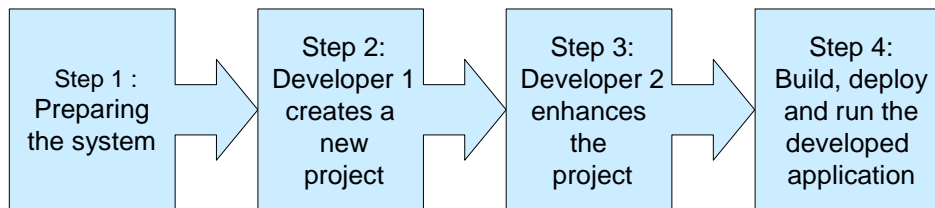
System, Installed Applications, and Authorizations

- Usage type AS Java is centrally installed. It is an obligatory prerequisite for installing usage type Development Infrastructure (that is, NWDI).
- SAP NetWeaver Development Infrastructure is centrally installed. It is needed only to provide the DTR. No other parts of the NWDI are used in the team-oriented development.
- SAP NetWeaver Developer Studio is installed on each developer's machine.
- You have permissions to set up the DTR configuration.
- You have permissions to create users in User Management Engine (UME).

Knowledge

- Basic understanding of the SAP NetWeaver Development Infrastructure (NWDI) concepts and development scenarios.
- Basic knowledge in application development.
- Knowledge of the DTR security roles and permissions.
- Basic knowledge in working with UME.

Process Flow

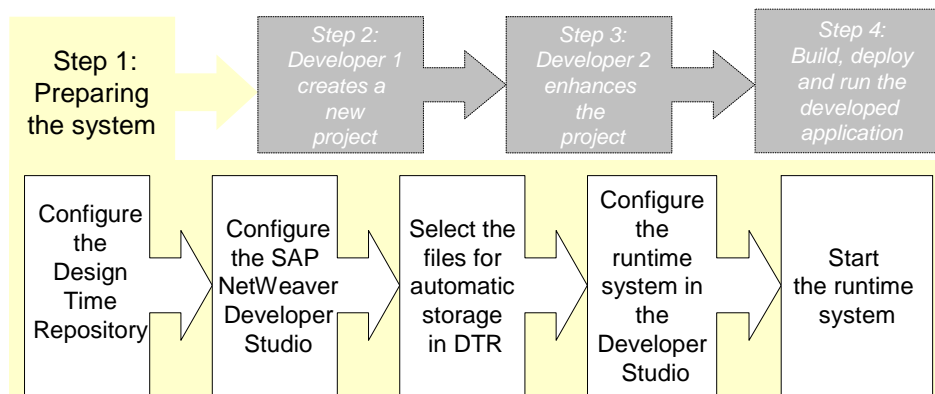


In this tutorial you take the following steps:

1. Preparing the system – prepare the system for team-oriented development.
2. Creating new projects – a developer creates a new project.
3. Enhancing the projects – another developer enhances the project that the first developer created.
4. Starting the application – the project is complete and ready to be exported to any customer.

1 Preparing the System

In this step, you set up the entire system for team-oriented development.



Prerequisites

Two machines are installed with a SAP NetWeaver Developer Studio and an AS Java on each.

Procedure

1. Configure the DTR Server

1. Configure the users and authorizations.
2. Create two users called `USER1` and `USER2` using the User Management Engine (UME).



To do this, you do not need the user `CMSadm` because the Change Management Service is not used in this scenario; you must create an *Admin*

315BGTranting Initial Privileges

user only if in the last step you want to grant separate authorizations for developers and administrators.

3. Create a hierarchy of workspace folders called **scenario1**. In it, create a workspace folder **tax**.
4. In the tax workspace folder, create a workspace **dev**.
5. (Optional) You can grant read and write permissions to both users in a new workspace – if you do not grant any permissions explicitly, the users have *all* rights.

2. Configure the DTR client in Developer Studio

Before a DTR repository can be used from inside the Developer Studio, you must perform some configuration steps:

1. In the Developer Studio open the *Design Time Repository* perspective.
From the menu path, choose *Window* → *Open Perspective* → *Other* → *Design Time Repository*.
2. Switch to the *Repository Browser* view. The browser view displays a single entry named *OFFLINE* or *No Clients Configured* to indicate that you are not yet connected to the DTR server.
3. Select *OFFLINE* and in the context menu, choose *Create Client*.
4. In the *New DTR Client* dialog that appears, specify **DTRClientOfUser1** as the client name.
5. Enter the URL of the installed DTR server in the form **http://<host>:<port>/dtr**.
Replace *<host>* and *<port>* with the corresponding values of your DTR server instance. For example, if the DTR server has been installed as *dtr-server.example.com* listening on port *50000*, enter *http://dtr-server.example.com:50000/dtr*.
6. Enter a local root directory that is used to replicate sources to your local machine. Replace the default value with **c:\DevInf**. Choose *OK*.
7. Repeat all these steps for the Developer Studio installed on the other's developer machine. Name the DTR client **DTRClientOfUser2**.

3. Select files for automatic storage in the DTR

You can use files that are already stored in the DTR if you have them or have created new files. For the purpose of the team-oriented development tutorial, you must change the default selected files for automatic storage in DTR:

1. In the Developer Studio, from the menu path, choose *Window* → *Preferences* → *Team* → *Ignored Resources*.
A list of file types appears. By default all types are selected, which means that they are not automatically added to activities and stored in the DTR.
2. If checked, uncheck the following file types:
 - a. *.project*
 - b. *.classpath*

When creating projects, the **unchecked** file types are added to activities to be stored in the DTR.

4. Configure the Runtime System in Developer Studio

315BGTranting Initial Privileges

1. In the Developer Studio, from the menu path choose *Window* → *Preferences* → *SAP AS Java*.
2. Select *SAP AS Java is installed on local host*. Choose *Browse*.
In the dialog that appears, select your local AS Java instance as *Local Engine Installation* and choose *OK*.
The system name, instance numbers and root paths of your engine are displayed in the *Preferences* dialog.
3. Choose *OK*.
4. Repeat all these steps for the Developer Studio installed on the other's developer machine.

5. Start the runtime system in the SAP NetWeaver Developer Studio

1. In the Developer Studio, from the menu path choose *Window* → *Show View* → *Server* → *Servers*.
2. In the root node of the engine, choose *Start Local Engine* to start the AS Java.



Note

If you no longer need the AS Java, you can stop the AS Java. To do that, choose *Stop Local Engine*.

1.1 Granting Initial Privileges

For the first authorizations, define two Access Control Lists (ACLs):

- One ACL for the root directory ("/")
This ACL contains the basic privileges for all users (read, change, check in).
- One ACL for the system directory
This ACL grants authorizations for administrators only.



Caution

The ACL for the root directory is essential: if you grant privileges only to inferior nodes, you lock the server for all users.

Procedure

1. Start the Design Time Repository perspective. In the repository browser, select the root node.



Note

This node is labeled `<user>` on `<system>`.

2. In the context menu, choose *View Permissions*.

The *Permissions* view of folder `/ws` appears.

315BG Granting Initial Privileges

3. To maintain authorizations for the root directory, enter its URL:

**Note**

The proposed value is the URL of the root directory (`http://<server>:<port>/dtr/`).

The *Permissions* view of the root directory appears.

4. In the *Permissions* view, choose *Add Principal* from the context menu.

A dialog window appears.

5. Make the following settings:

1. As the *Principal Type*, select *ALL*.

2. Select the following principles:

- *access*

- *read*

3. Confirm your settings with *OK*.

6. In the *Repository Browser*, navigate to the directory */ws/system*.

7. In the context menu, choose *View Permissions*.

The *Permissions* view for */ws/system* appears.

8. In the *Permissions* view, choose *Add Principal* from the context menu.

A dialog window appears.

9. Make the following settings:

1. As the *Principal Type*, select *ALL*.

2. Select the following privileges:

- *access*

- *read*

- *write*

3. As the *Privilege Type*, select *deny*.

4. Confirm your settings with *OK*.

10. Again select *Add Principal*.

A dialog window appears.

11. Make the following settings:

1. As the *Principal Type*, select *USER*.

2. Enter the name of the administrator.

313B Creating Users and Assigning Groups Using UME

3. Select the following privileges:
 - *access*
 - *read*
 - *write*
 4. Confirm your settings with *OK*.
12. Repeat these steps for all administrators.
13. To activate your changes, choose *Activate all Changes*. Open a browser window under the URL `http://<DTR server>:<port>/dtr/sysconfig/support/AclRefresh` and choose *Refresh*.

Emergency User Account

You have set the permissions too restrictively or made invalid specifications for users or groups when creating ACLs; as a result, you have denied access to all users or at least to yourself. This implies that you do not have the authorization to undo these changes. In this case, you can use a special user account with all rights to all resources. The default name of this user account is *superadmin*.

To be able to use this emergency user, in the User Management Engine create a user *superadmin* in your system. You can then use this user to log on and access all resources.



Note

You can change the name of the emergency user.

1.2 Creating Users and Assigning Groups Using UME

In this step, you use the User Management Engine (UME) as the user store.

Procedure

You must create users *USER1*, *USER2* in the UME of the AS Java of the NWDI.

1. Using the UME, create the following users.
 - **USER1** and **USER2**

These users represent the developers in the SAP NetWeaver Developer Studio and in the NWDI, and need access to the SLD and to all components of the NWDI.
 - **Admin**

This administrator of the NWDI needs access to all components of the NWDI. You do not need access to the SLD to configure the CMS or to operate the NWDI. (SLD tasks such as creating software components and namespace prefixes can be done by an SLD administrator.)

313BCreating Users and Assigning Groups Using UME



The AS Java requires that you change the initial password of a user at the first logon. Therefore, after creating the users, log on to the AS Java once with every user and change the initial password.

If the SLD is installed on a different AS Java, make sure that you use identical passwords in all systems.

- Assign the users you created to the following groups.

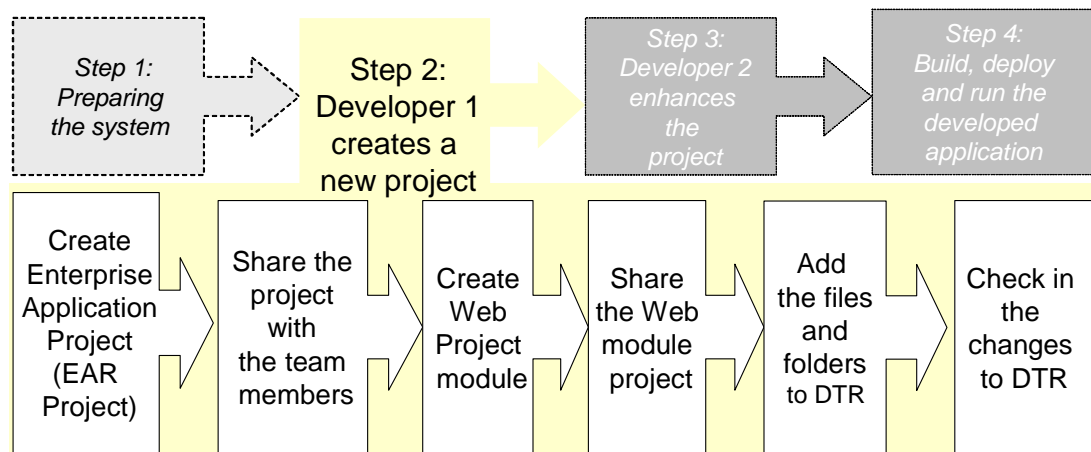
User	Group
USER1	NWDI.Developers
USER2	NWDI.Developers
Admin	NWDI.Administrators

Result

You have created the required users for the NWDI and assigned them to the required groups.

2 Creating New Projects

In this step, the first developer (called *Developer1*) creates a new project that will be used for the development in the team. The application that is developed in this tutorial shows a Web page for tax calculation: you enter an income and the tax amount is displayed. In our example, *Developer1* is the project leader and he or she creates the projects in the first step of the development of the application.



These are the steps *Developer1* takes using the SAP NetWeaver Developer Studio:

- Share the project with the team members.
- Create a Web module project.
- Share the Web module project.
- Add the files and folders to DTR.
- Check in the entire project to Design Time Repository (DTR)

Procedure

1. Create an Enterprise Application Project

1. From the menu path of the Developer Studio, choose **File** → **New** → **Project** → **J2EE** → **Enterprise Application Project**. Choose **Next**.
2. Name the new project `TaxEAR`. Choose **Finish**.

The Java EE perspective is opened and the Project Explorer view becomes visible. An entry for your new project is created in the Explorer.

2. Share the Project with Team Members

You use the DTR to share projects with all team members. Every team member must have access to the DTR.

1. Log on to the DTR if you have not already done so.
 1. Open the Design Time Repository perspective and select your DTR client, that is, `DTRClientOfUser1` from **DTR** → **Select Client** in the menu path.
 2. Open the context menu of the root node in the Repository Browser view and choose **Login**.

A dialog *Logon to Development Infrastructure* appears.

3. Enter `USER1` and the corresponding password and choose **Log on**.
2. In the Java EE perspective, switch to the Navigator view.

From the menu patch, choose **Window** → **Show View** → **Navigator**.
3. In the project tree, select the entry `TaxEAR`. In the context menu, choose **Team** → **Share Project**...

The *Share Project* dialog appears.

1. Select Design Time Repository as the repository type. Choose **Next**.

The upper part of the dialog shows the folder structure on the DTR server.
2. Browse to `scenario1/tax/dev` in this structure and select `dev`.
3. Choose **Create Folder** and enter `projects/TaxEAR`. Choose **OK**.
4. Select the `TaxEAR` folder and choose **Finish**.

The project together with its content is moved to the folder `c:\DevIn\scenario1\tax\dev\projects\TaxEAR`.



Note

This is the location to which the Developer Studio later on replicates the sources from the DTR server.

4. In the Navigator view, select the project `TaxEAR`. From the context menu, choose **DTR** → **Add** → **Subtree**.

313BCreating Users and Assigning Groups Using UME

The dialog window *Add local-only Resources* appears.

5. Accept the proposed resources by choosing *OK*.

A dialog box *Select Activity* appears.

6. To create a new activity, in the dialog choose *New*.



Note

The activity is used to record your changes. You can add several changes (which should belong together) to a single activity.

7. Enter *New Projects* as the display name of the new activity. Choose *OK*.

A dialog box appears.

8. In the *Select Activity* dialog, select the new activity and choose *OK*.

The files of your project are added to the activity. They are not yet checked in to DTR, so you can use the same activity to create some other projects.

3. Create a Web Module Project

1. To create a Web project, from the menu path choose **File** → *New* → *Project* → *Web* → *Dynamic Web Project*.
2. Name the new project `TaxWeb` and accept the default project location. Choose *Finish*.

The Web project is created.

4. Share the Web Module Project

1. Open the *Navigator* view.
2. Select the entry *TaxWeb* in the project tree. Choose **Team** → *Share Project...* from the context menu.

A dialog window *Share Project* appears. Choose *Design Time Repository* as the repository type and choose *Next*.

3. Select the remote folder `. scenario1/tax/dev/projects`
4. Choose *Create Folder* and enter `TaxWeb`. Choose *OK*.
5. Select the *TaxWeb* folder and choose *Finish*.

The project together with its content is moved to the `c:\DevInf\scenario1\tax\dev\projects\TaxWeb` folder.

6. In the *Navigator* view, select the project *TaxWeb* and expand the project node.

5. Add Files and Folders to the DTR

1. To add subfolders in the project folder to the DTR:

313B Creating Users and Assigning Groups Using UME

1. Select all subfolders of the project folder except the *bin* folder. With multi-select, you can select either files or folders.



Note

The SAP NetWeaver Developer Studio uses the *bin* folder for compiling your projects. For other technologies (WebDynpro, Data Dictionary and so on) you may have to select another set of folders here.

2. From the context menu, choose **DTR** → **Add** → **Folder**.
 3. In the dialog that appears, select activity *New Projects* and choose *OK*.
2. To add files in the project folder to the DTR:
1. Select all files that are located directly in the project folder.
 2. From the context menu, choose **DTR** → **Add**.
 3. In the dialog that appears select activity *New Projects* and choose *OK*.

6. Check in the Changes to DTR

1. Open the *Open Activities* view.

Choose **Window** → **Show View** → **Other** → **Design Time Repository** → **Open Activities**.

2. Select the activity *New Projects*.
3. Choose *Checkin* from the context menu.
4. In the dialog that appears, choose *OK*.

Your changes now are written to the DTR.

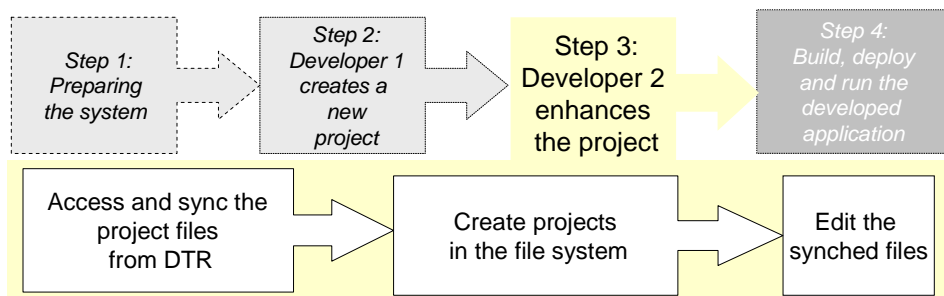


Note

The activity is now closed. For further changes, you have to create a new activity.

3 Enhancing the Projects

In this step, a second developer (*USER2*) adds code to the newly created project.



Procedure

1. Access and sync project files in the DTR

3. Open the Design Time Repository perspective.
From the menu path of the SAP NetWeaver Developer Studio, choose *Window* → *Open Perspective* → *Other* → *Design Time Repository*.
4. To log on to the DTR, choose the *Log On...* button in the toolbar.
5. In the dialog that appears, enter **USER2** and the corresponding password.
In the *Repository Browser* view you will see the folder structure on the DTR server.
6. Browse the structure and locate workspace *scenario1/tax/dev/projects*.
7. Select the *projects* folder and choose *Sync* from the context menu.
This replicates the projects *TaxEAR* and *TaxWeb* to your local file system.

2. Create projects in your file system

You need projects to be able to edit the files.

1. Select folder *TaxEAR*.
2. Choose *Create Project* from the context menu.
3. Accept the default project name. Choose *OK*.
4. Select folder *TaxWeb*.
5. Choose *Create Project* from the context menu.
6. Accept the default project name. Choose *OK*.

3. Edit files synced from the DTR

1. Open the *J2EE* perspective and switch to the *Project Explorer* view.
2. Select the *TaxWeb* project and choose *New* → *JSP* from the context menu.
3. In the dialog that appears, enter the name **tax** and choose *OK*.
4. In the dialog that appears, to create a new activity for your changes, choose *New*. Enter **Add Content** as the display name of the new activity.
5. Enter a description of the activity and choose *OK*.
6. In the dialog that appears, select the new activity and choose *OK*. Confirm the file selection with *OK*.
7. Again select the activity *Add Content*.
The file *tax.jsp* now is attached to the activity but it is not yet checked in.
The JSP source appears.
8. Switch to the *Source* pane and enter the following code:

```
<%@ page language="java"%>
<html>
<head>
<title>Tax Calculator</title>
</head>
<body bgcolor="#c8e3ff">
```

313BCreating Users and Assigning Groups Using UME

```

<h1>Simple Tax Calculator</h1>
<form action="tax.jsp" method="POST">
<table border="0" cellspacing="5" cellpadding="5">
  <colgroup>
    <col width="20%" />
    <col width="80%" />
  </colgroup>
  <%
    double income;
    try {
      income =
Double.parseDouble(request.getParameter("income"));
    } catch (Exception e) {
      income = 0;
    }
  %>
  <tr>
    <td>Income:</td>
    <td><input name="income" value="<%= income %>" type="text"
      size="30" maxlength="40"></td>
  </tr>
  <tr>
    <td>Tax:</td>
    <td><%= (income * 0.3) %></td>
  </tr>
  <tr>
    <td colspan="2" align="center"><input type="submit"
      value="calculate tax"></td>
  </tr>
</table>
</form>
</body>
</html>

```

9. Save your entries.
10. In the *Project Explorer* view, select the *TaxEAR* project. Choose *Modules* and from the context menu, choose *Add/Remove*.
11. In the dialog that appears, choose the *TaxWeb* project and choose *OK*.
12. Select the *tax.jsp* file and from the context menu, choose *DTR* → *Edit*.

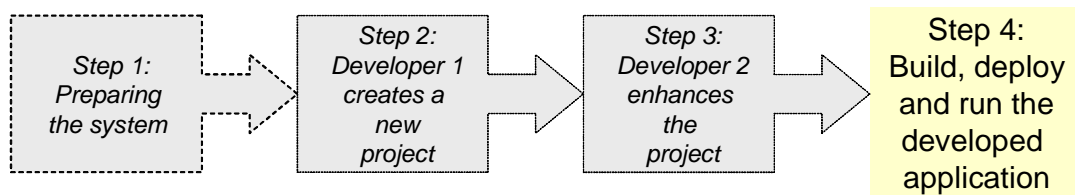
The file is checked out for editing.



Usually the activities are checked in only after the sources are built and tested. After they are checked in, the versions contained in them are available for the team on the server.

4 Starting the Application

In this final step you build, deploy and run the example application.



Procedure

1. Build and deploy the project

1. Open the *Servers* view and select the AS Java server on which you want to deploy the application.
2. From the context menu, choose *Add and Remove Projects*.
3. Select the *TaxEAR* project. Choose *Add* and then *Finish*.

The EAR file is built and deployed automatically to the AS Java.

4. Test the JSP using the address
`http://<AS Java host>:<AS Java port number>/tax.jsp`.

2. Check in your changes to the DTR

After the successful test, save your data on the DTR server.

1. Open the Design Time Repository perspective.
2. In the *Open Activities* view locate the activity *Add Content* and in the context menu choose *Checkin*.
3. In the *Checkin* dialog, enter a description for the activity. Choose *OK*.

The activity is checked in. It is now in *closed* state and you can no longer change it. The versions in this activity are now available on the DTR server.

Result

Your tax calculation is executable. The sources are stored on the DTR server.

This completes this tutorial.