PRINTABLE VERSION

Importing Eclipse Projects into NetBeans IDE

REQUIRES NetBeans 7.4/7.3/7.2

Contents

To complete this tutorial, you need the following software and resources.

Software or Resource	Version Required
NetBeans IDE	version 6.9, 7.0, 7.1, 7.2, 7.3, or 7.4
Java Development Kit (JDK)	version 6 or 7

The Import Eclipse Project Wizard

An enhanced support for importing Eclipse projects is available in the NetBeans IDE. The Import Eclipse Project wizard enables you to use NetBeans to work on Java desktop and web application projects that were created in Eclipse and MyEclipse.

After creating a NetBeans project from an Eclipse project, you can work in the following ways:

- Use NetBeans on a team project where other developers are using Eclipse.
- Use NetBeans and Eclipse interchangeably on a project.
- Try out NetBeans on an existing project that was set up in Eclipse.

top

How the Import Wizard Works

When you import a project, the IDE analyzes the Eclipse project metadata and then creates a NetBeans project based on that metadata. The project import does

Download NetBeans IDE

Training

- Java Programming Language

Support

- Oracle Development Tools Support Offering for NetBeans IDE

Documentation

- General Java Development
- External Tools and Services
- Java GUI Applications
- Java EE & Java Web Development
- Web Services Applications
- NetBeans Platform (RCP) and Module Development
- PHP and HTML5 Applications
- C/C++ Applications
- Mobile Applications

- Sample Applications
- Demos and Screencasts

More

- FAQs
- Contribute Documentation!
- Docs for Earlier Releases

not impose changes on the Eclipse project structure, so it does not interfere with the way the project works in Eclipse.

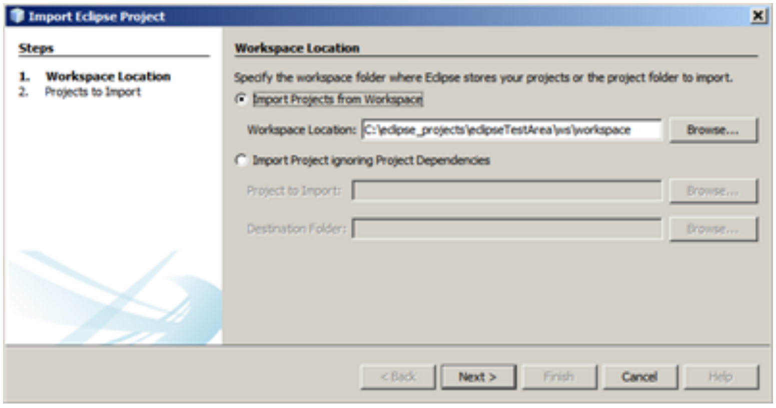
Once you have the project imported, you can use either or both the NetBeans and Eclipse IDEs to edit, build, run, debug, and profile the project. If you make a configuration change in the Eclipse project, you can resynchronize the NetBeans project. This enables you to work on projects in NetBeans even if everybody else on your team is working with Eclipse. The build path that you set in Eclipse remains the master build path for the project.

top

Importing and Configuring Projects

To import an Eclipse project into the IDE:

1. Make sure that you have copy of the project that you want to import on your system.
   Typically, this project would already be in an Eclipse workspace on your system.
2. Choose File > Import Project > Eclipse Project.
3. In the Workspace Location page of the wizard, select the Import Projects from Workspace radio button and specify the workspace location. Click Next.



Specify the workspace folder where Eclipse stores your projects or the project folder to import.

● Import Projects from Workspace

Workspace Location: C:\eclipse_projects\eclipseTestArea\ws\workspace

○ Import Project ignoring Project Dependencies

Project to Import:

Destination Folder:

4. In the Projects to Import page, select the projects that you want to import.
5. Choose one of the following two options for storing the NetBeans project files:
   - **Store NetBeans project data inside Eclipse project folders.** NetBeans adds folders and files within the top-level folder of the original project.
   - **Create imported NetBeans projects in a separate location.** NetBeans uses the sources and libraries in the original Eclipse project folder but creates a separate folder to hold NetBeans project metadata and build outputs.

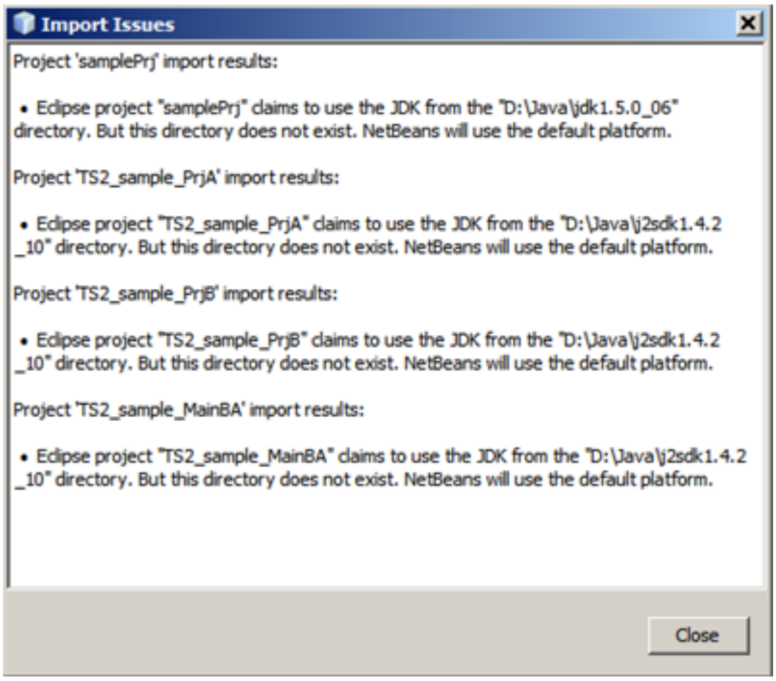
Note: Typically, it is better to store NetBeans project data inside Eclipse project folders. In most cases, this means that the NetBeans project metadata will refer to sources and libraries with the same

paths that are used by the Eclipse metadata. Therefore, checking out the project from a version control system on different machines should result in similar behavior both in NetBeans and Eclipse.
See NetBeans Project Files below for a list of files and folders that the NetBeans IDE creates.

6. (Applicable when web applications are being imported only.) Click Next. In the Servers page, register any servers that your projects need with NetBeans IDE.
7. Click Finish.

After you have completed the wizard, the following dialog boxes might appear:

- **Import Issues.** This dialog box provides information about discrepancies between the project structure in Eclipse and in NetBeans and points out actions that you might need to take to correct the discrepancies.
  You can copy the information from this dialog and paste it elsewhere for future reference.
  In most cases, you use the project's Project Properties dialog box to resolve those issues. See Resolving Import Problems for a guide to resolving the most common problems.



Project 'samplePrj' import results:

- Eclipse project "samplePrj" claims to use the JDK from the "D:\Java\jdk1.5.0_06" directory. But this directory does not exist. NetBeans will use the default platform.

Project 'TS2_sample_PrjA' import results:

- Eclipse project "TS2_sample_PrjA" claims to use the JDK from the "D:\Java\j2sdk1.4.2_10" directory. But this directory does not exist. NetBeans will use the default platform.

Project 'TS2_sample_PrjB' import results:

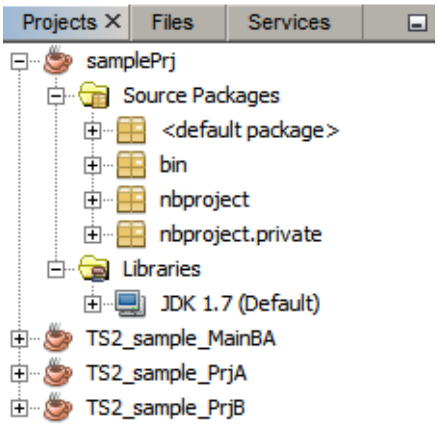
- Eclipse project "TS2_sample_PrjB" claims to use the JDK from the "D:\Java\j2sdk1.4.2_10" directory. But this directory does not exist. NetBeans will use the default platform.

Project 'TS2_sample_MainBA' import results:

- Eclipse project "TS2_sample_MainBA" claims to use the JDK from the "D:\Java\j2sdk1.4.2_10" directory. But this directory does not exist. NetBeans will use the default platform.

- **Resolve Reference Problems.** This dialog box alerts you to a specific reference problem with one of your project libraries. You can solve this problem after dismissing this dialog box by right-clicking the project's node and choosing Resolve Reference Problems.
- **Resolve Missing Server.** This dialog box alerts you that the project can not find a necessary server. You can solve this problem after dismissing this dialog box by right-clicking the project's node and choosing Resolve Missing Server.

After you have completed the wizard and have closed any of the above informational dialog boxes, nodes for the projects will appear in the Projects window.



If there are references in your project metadata to servers or other resources that NetBeans can not resolve, the node for the project will appear in red. You can resolve these references immediately by right-clicking the project node and choosing Resolve Reference Problems or Resolve Missing Server.

For other types of project configuration adjustments, you use the Project Properties dialog box. You open the Project Properties dialog box by right-clicking the project's node and choosing Properties.

For more information on configuring your project, see Configuring the Classpath and Other Project Settings in the Creating, Importing, and Configuring Java Projects guide.

top

Building and Running a Project

Once you have the project imported into NetBeans, you can build and run the project. All artifacts created from NetBeans build and run commands are created in the `build` and `dist` folders. NetBeans does not over-write output created from Eclipse build actions. If the Eclipse project already has `build` and `dist` folders, the NetBeans project creates folders called `nbbuild` and `nbdist` and uses those for the build outputs.

The following are some of the build and run commands are available from the Run menu:

- **Run Project.** Test runs the application in the IDE.
- **Clean and Build Project.** Deletes the contents of the `build` (or `nbbuild`) and `dist` (or `nbdist`) folders and rebuilds all of the project's outputs. Uses the NetBeans build script. Similar to the Clean command in Eclipse.
- **Clean.** Deletes the contents of the `nbbuild` and `nbdist` folders.
- **Build.** Rebuilds the project's outputs. If the Compile on Save feature is enabled, the Build command is disabled. See the Compile on Save section of the Creating, Importing, and Configuring Java Projects guide.

For more information on building and running your application as well as customizing the build process, see Creating, Importing, and Configuring Java Projects.

top

NetBeans Project Files

After you have imported the project, you will find the following folder and files on your system:

- `build.xml` file or `nb-build.xml` file. The main NetBeans build script for the project. You can customize this script according to the needs of your project. By default, this file is called `build.xml`. If such a file already exists in the project folder, the script is called `nb-build.xml`.
- `nbproject` folder. Contains most of the NetBeans project metadata, including resources that are called by the main NetBeans build script. If you check this folder and the `build.xml` file or `nb-build.xml` into your version control system, other users will be able to open the project in NetBeans. This folder also contains the `private` folder, which contains data specific to your system. This folder should not be checked in to the version control system since its contents will vary between users. See Version Control Considerations below.
- `nbbuild` folder. When you build or run your project in NetBeans, the project's sources are compiled into this folder.
- `nbdist` folder. When you build your project in NetBeans, the project's distributable outputs are created and placed in this folder. Such outputs might be JAR files and WAR files.

top

Version Control Considerations

If the project is checked out of a version control system, the `build` (or `nbbuild`), `dist` (or `nbdist`), and the `nbproject/private` folders should not be checked into that version control system.

If the project is under the CVS, Subversion, or Mercurial version control systems, the appropriate "ignore" files are created or updated for these directories when the project is imported.

Though `nbproject/private` should be ignored, `nbproject` should be checked into the version control system. `nbproject` contains project metadata that enables others users to open the project in NetBeans without having to import the project first.

top

Resynchronizing a Project

The project importer features synchronization capabilities. If the classpath in the Eclipse has changed since you initially imported it, you can use the Resynchronize Eclipse Projects feature to update the classpath in the corresponding NetBeans project.

Project resynchronization is one-way from Eclipse projects to NetBeans projects. If you make changes to the project structure in NetBeans, those changes are not propagated to the Eclipse project with the resynchronization feature. If you intend to keep both Eclipse and NetBeans projects, use the Eclipse project as the "master" project.

The IDE also resynchronizes the projects automatically if the changes to the Eclipse configuration are unambiguous and do not require your input. This automatic resynchronization occurs shortly after you open the project. If the resynchronization requires your input, you need to manually resynchronize the project.

To manually resynchronize NetBeans projects with Eclipse projects:

- Choose File > Import Project > Resynchronize Eclipse Projects.

Notes: When you resynchronize a project, the resynchronization is performed on all projects that you have imported from the workspace.

top

Resolving Import Problems

When you import a project into NetBeans, there might be some things that can not be automatically resolved in NetBeans IDE. For some of these problems, a menu item, such as Resolve Missing Server Problem, appears in the contextual menu for the project. Other problems can be resolved in the Project Properties dialog box for the imported project in NetBeans IDE.

Here is a list of common import problems and their solutions.

Problem Message	Solution
Resolve Missing Server Problem	Right-click the project node and choose Resolve Missing Server Problem. Then navigate to the file or folder that contains the server.
Resolve Reference Problem	Right-click the project's node and choose Resolve Reference Problem. Then navigate to the file or folder that contains the resource that is referred to from the project.
Eclipse platform for project *ProjectName* cannot be used. It is a JRE and the NetBeans project requires a JDK. NetBeans will use the default platform.	If you would like to change the platform that NetBeans uses for the project, choose Tools > Platforms and specify a different platform.
Eclipse project *ProjectName* claims to use JDK from the "{1}" directory. But this directory does not exist. NetBeans will use the default platform.	If you would like to change the platform that NetBeans uses for the project, choose Tools > Platforms and specify a different platform.

NetBeans does not support source includes/excludes per source root as Eclipse does. They were merged and it is recommended that you double check them in project's properties in Source panel.	In NetBeans, includes and excludes are declared in one place for the whole project. To check the includes and excludes in the NetBeans project, right-click the project's node and the Projects window and choose Properties. In the Project Properties dialog box, select the Sources tab and then click the Includes/Excludes button.
Import failed due to .... More details can be found in IDE's log file.	You can open the IDE's log file by choosing View > IDE Log.
Unknown project type - it cannot be imported.	You can only import the following Eclipse project types: Java Project, Java Project from Existing Ant File, Static Web, Dynamic Web, and JPA Project.

top

Send Feedback on This Tutorial

See Also

- Importing an Eclipse or JBuilder Project in *Developing Applications with NetBeans IDE*
- NetBeans For Eclipse Users
- Creating, Importing, and Configuring Java Projects
- General Java Development Learning Trail
- Java GUI Applications Learning Trail
- Java EE & Java Web Learning Trail

top

SiteMap About Us Contact
Legal & Licences