Eclipse

1. Liferay7 Portlet example:

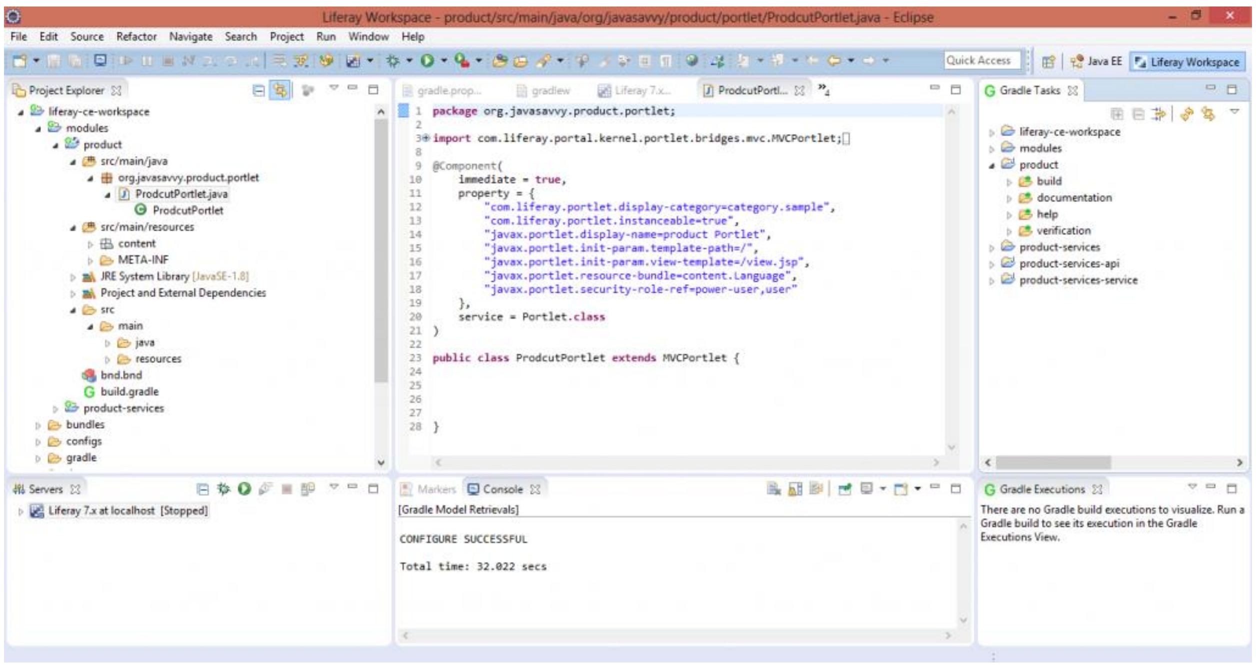
http://www.javasavvy.com/liferay-7-portlet-tutorial/
https://www.opensource-techblog.com/complete-liferay-guide

We just created a portlet module with name "product" and folder structure will look like below:

- All web resources in Liferay 7 are created under src/main/resources folder but where as Liferay 6 creates in docroot/html folder
- Portlet configuration such as portlet.xml and liferay-portlet.xml configuration need to updated in @Component annotation under property attribute
- The rest of Portlet API sush doView, processAction and servceResource all are same and you can directly use the API in JSP's and Portlet Controller

@Component
all portlet configuration goes into @**Component** annotation that used to deploy in OSGI container
immediate – module will be deployed and should be started immediately instead of being lazy-loaded
property –   All java portlet properties should be prefixed by: ***javax.portlet*** Lifera y properties are prefixed by: ***com.liferay.portlet.***
portlet.xml and liferay-portlet.xml   OSGI mappings :   you can find list of mappings on liferay site : ***https://dev.liferay.com/develop/reference/-/knowledge_base/7-0/portlet-descriptor-to-osgi-service-property-map***

```
@Component(
        immediate = true,
        property = {
       "com.liferay.portlet.display-category=category.sample",
       "com.liferay.portlet.instanceable=true",
       "javax.portlet.display-name=product Portlet",
        "javax.portlet.init-param.template-path=/",
        "javax.portlet.init-param.view-template=/view.jsp",
        "javax.portlet.resource-bundle=content.Language",
        "javax.portlet.security-role-ref=power-user,user"
         },
       service = Portlet.class
)
```



Liferay Workspace - product/src/main/java/org/javasavvy/product/portlet/ProdcutPortlet.java - Eclipse

```
package org.javasavvy.product.portlet;

import com.liferay.portal.kernel.portlet.bridges.mvc.MVCPortlet;

@Component(
    immediate = true,
    property = {
        "com.liferay.portlet.display-category=category.sample",
        "com.liferay.portlet.instanceable=true",
        "javax.portlet.display-name=product Portlet",
        "javax.portlet.init-param.template-path=/",
        "javax.portlet.init-param.view-template=/view.jsp",
        "javax.portlet.resource-bundle=content.Language",
        "javax.portlet.security-role-ref=power-user,user"
    },
    service = Portlet.class
)

public class ProdcutPortlet extends MVCPortlet {




}
```

```
[Gradle Model Retrievals]

CONFIGURE SUCCESSFUL

Total time: 32.022 secs
```

There are no Gradle build executions to visualize. Run a Gradle build to see its execution in the Gradle Executions View.

Liferay 7 Portlet tutorial and gradle task

Liferay provides bnd.bnd and build.gradle files to manage build and deployment process

bnd.bnd : bnd file is MANIFEST configuration template file which used to auto create MANIFEST.MF, so developers are not required to focus on MANIFEST file creation

It also used to activate Bundle with BundleActivator configuration if we are not going to use OSGI Declarative Services.

Bundle-SymbolicName: product

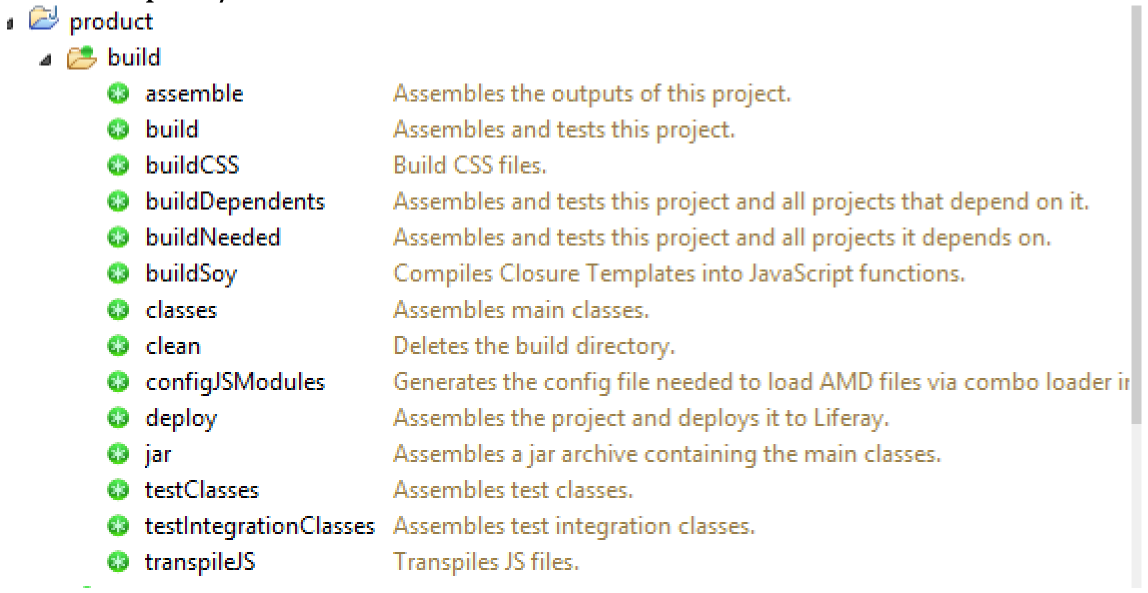
Bundle-Version: 1.0.0

build.gradle: build.gradle is used for dependency and build management for Liferay 7 modules.

Liferay 7 provides Gradle Build Task panel to execute build,deploy, clean, buildService task

The generated build artifacts can be copied to deploy or /bundles/osgi/modules folder for manual deployment

For auto deployment, make sure that tomcat is created under liferay-workspace/bundle folder



```
product
  build
    assemble                 Assembles the outputs of this project.
    build                    Assembles and tests this project.
    buildCSS                 Build CSS files.
    buildDependents          Assembles and tests this project and all projects that depend on it.
    buildNeeded              Assembles and tests this project and all projects it depends on.
    buildSoy                 Compiles Closure Templates into JavaScript functions.
    classes                  Assembles main classes.
    clean                    Deletes the build directory.
    configJSModules          Generates the config file needed to load AMD files via combo loader ir
    deploy                   Assembles the project and deploys it to Liferay.
    jar                      Assembles a jar archive containing the main classes.
    testClasses              Assembles test classes.
    testIntegrationClasses   Assembles test integration classes.
    transpileJS              Transpiles JS files.
```

Liferay Gradle Tasks

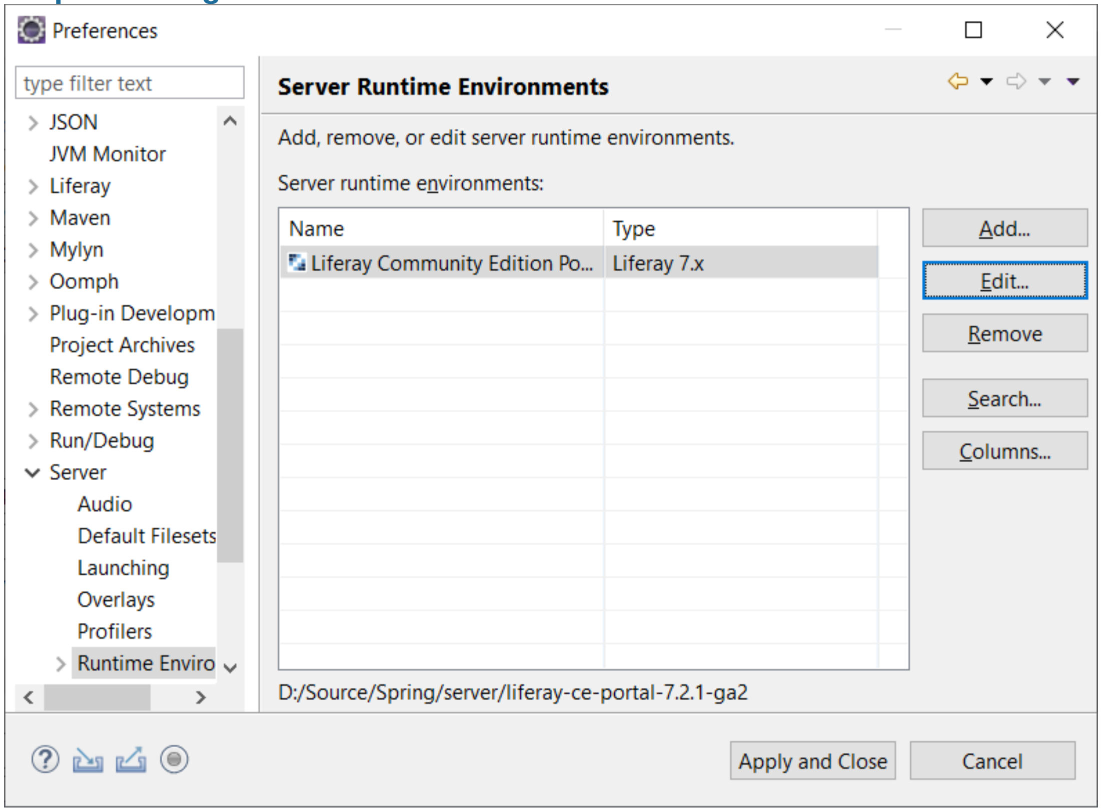
click on “deploy” target and you can see the logs that module is stated. The rest of portlet development is same as Liferay 6.

Now logon into Liferay portlet with test@liferay.com and add the portlet to page. that’s all .. you have developed Liferay7 Portlet.

https://portal.liferay.dev/docs/7-0/tutorials/-/knowledge_base/t/installing-liferay-ide

2. Liferay 7 development setup

Step 1: Config server



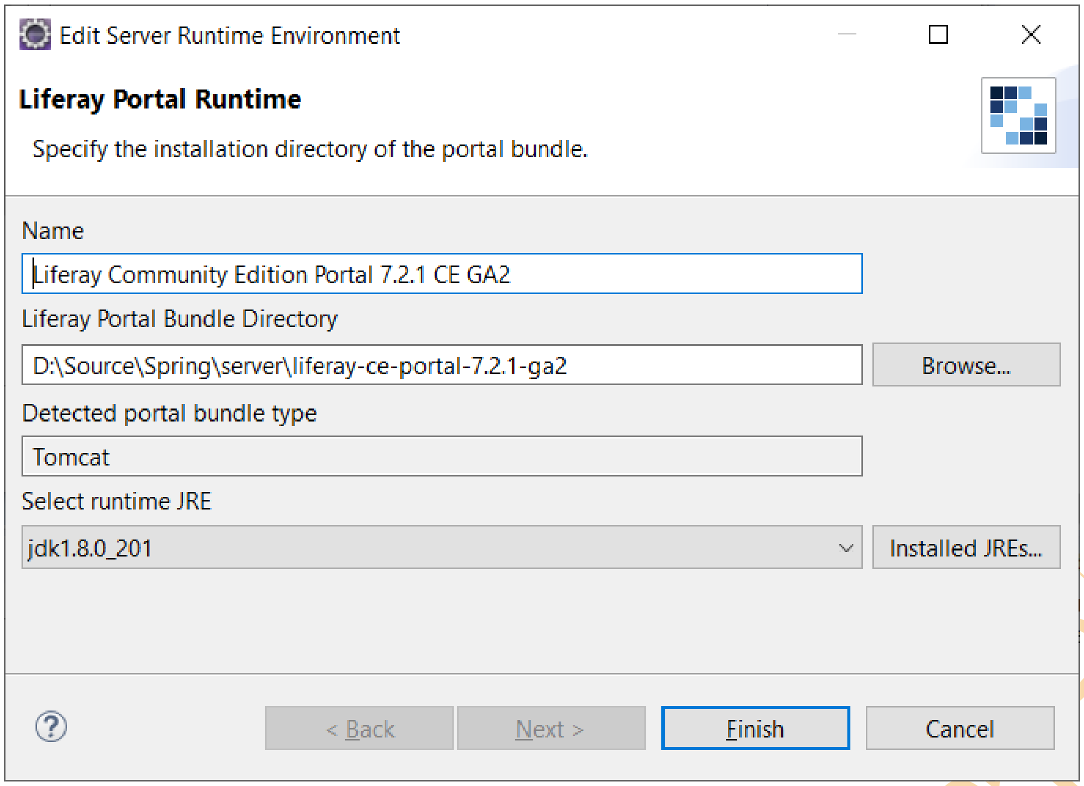
Server Runtime Environments

Add, remove, or edit server runtime environments.

Server runtime environments:

Name	Type
Liferay Community Edition Po...	Liferay 7.x

D:/Source/Spring/server/liferay-ce-portal-7.2.1-ga2



Liferay Portal Runtime

Specify the installation directory of the portal bundle.

Name: Liferay Community Edition Portal 7.2.1 CE GA2

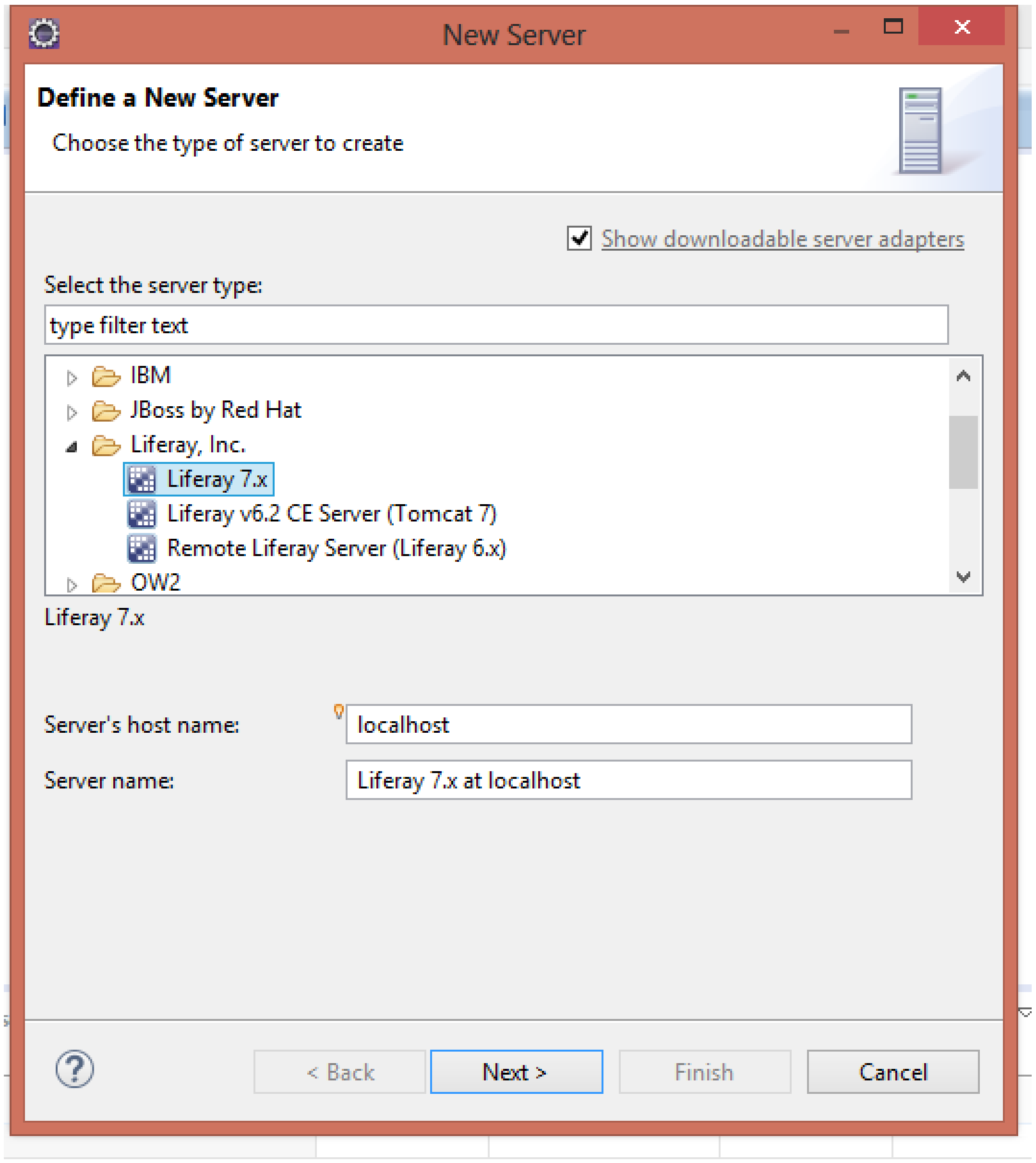
Liferay Portal Bundle Directory: D:\Source\Spring\server\liferay-ce-portal-7.2.1-ga2

Detected portal bundle type: Tomcat

Select runtime JRE: jdk1.8.0_201

Step 2: Liferay 7 Tomcat Setup in Eclipse

Now, we will configure tomcat in Liferay. Click on **File->New-> Server;** Select Liferay 7.x and click on **Next**. Update the tomcat server location with download tomcat and Lieray bundles directory with in liferay home(which you extracted liferay-ce-portal-tomcat-7.0-ga3)

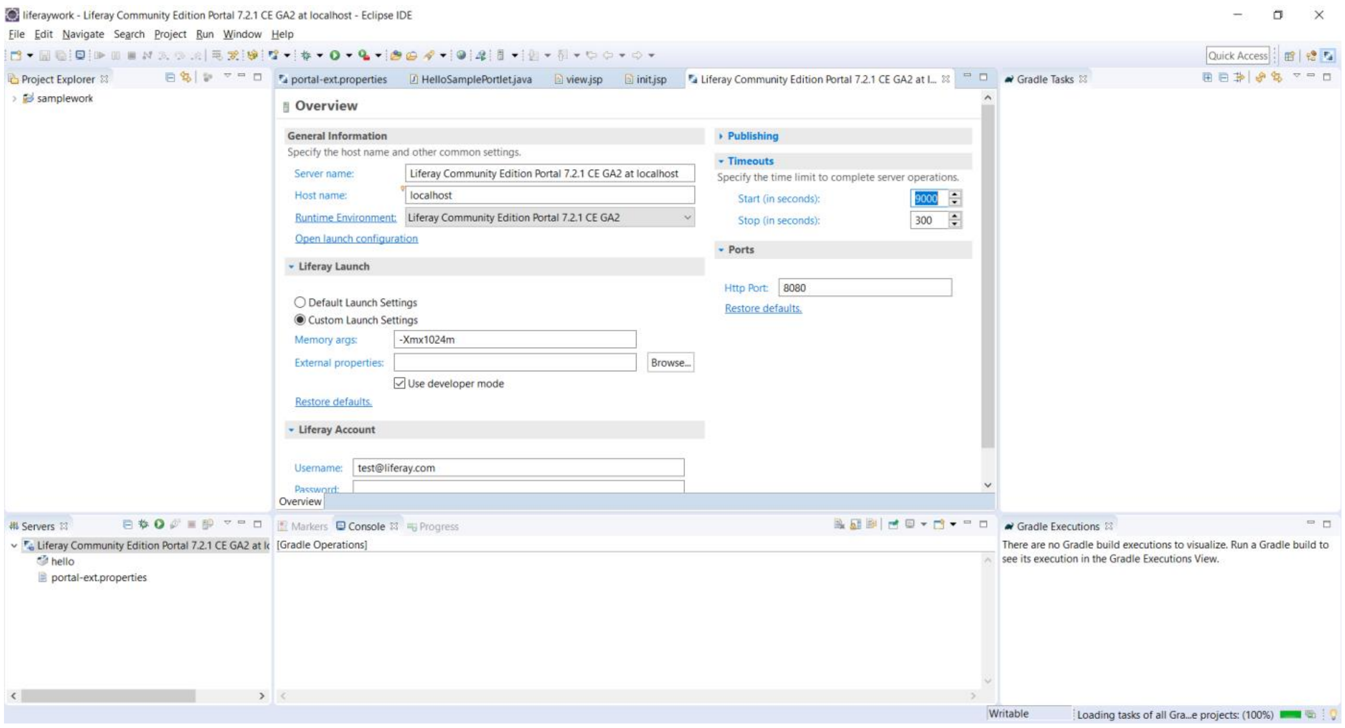


Liferay 7.x

Server's host name: localhost

Server name: Liferay 7.x at localhost

Set timeouts start = 9000



liferaywork - Liferay Community Edition Portal 7.2.1 CE GA2 at localhost - Eclipse IDE

Overview

General Information

Specify the host name and other common settings.

Server name: Liferay Community Edition Portal 7.2.1 CE GA2 at localhost

Host name: localhost

Runtime Environment: Liferay Community Edition Portal 7.2.1 CE GA2

Open launch configuration

Liferay Launch

Default Launch Settings [✓ Custom Launch Settings]

Memory args: -Xmx1024m

External properties:

[✓] Use developer mode

Restore defaults.

Liferay Account

Username: test@liferay.com

Password:

Publishing

Timeouts

Specify the time limit to complete server operations.

Start (in seconds): 9000

Stop (in seconds): 300

Ports

Http Port: 8080

Restore defaults.

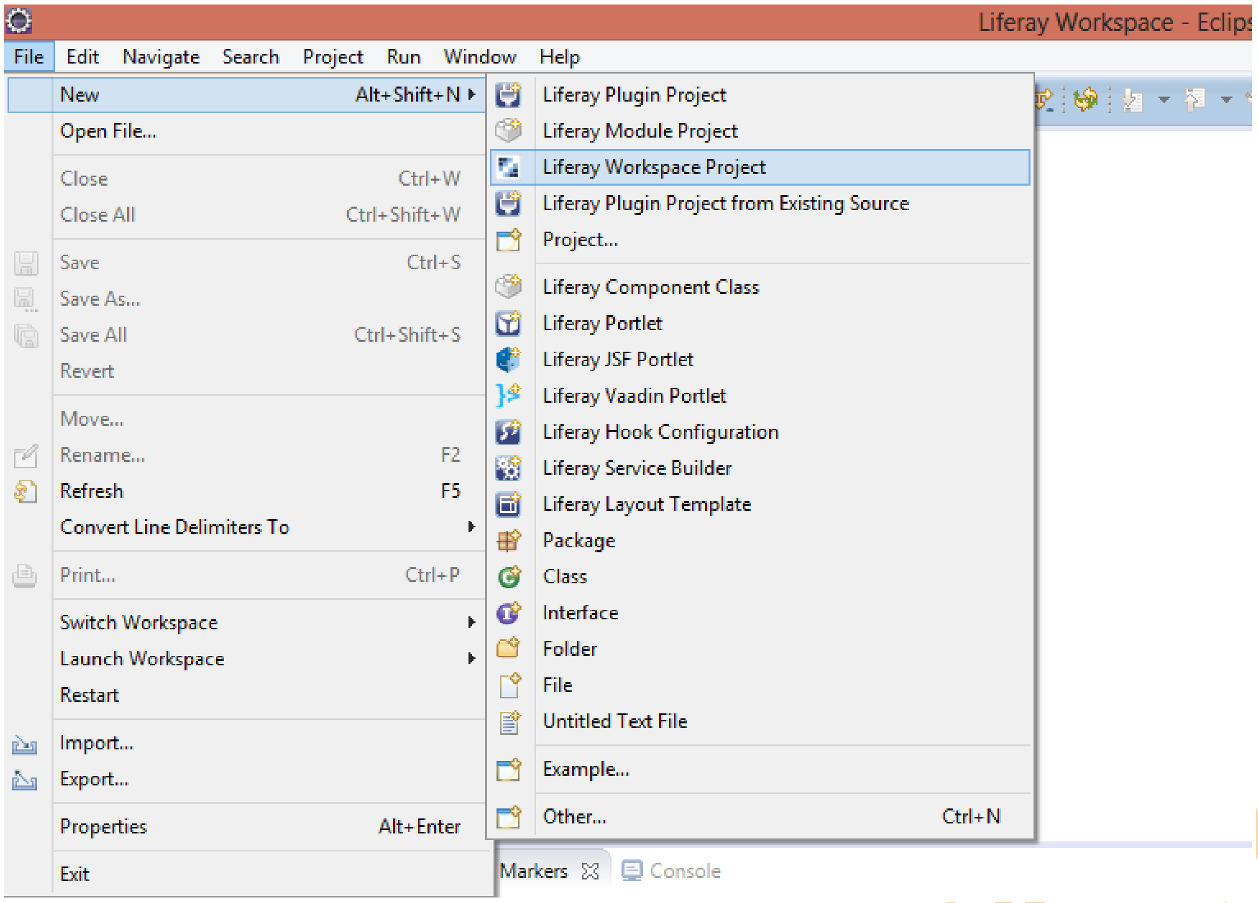
[Gradle Operations]

There are no Gradle build executions to visualize. Run a Gradle build to see its execution in the Gradle Executions View.

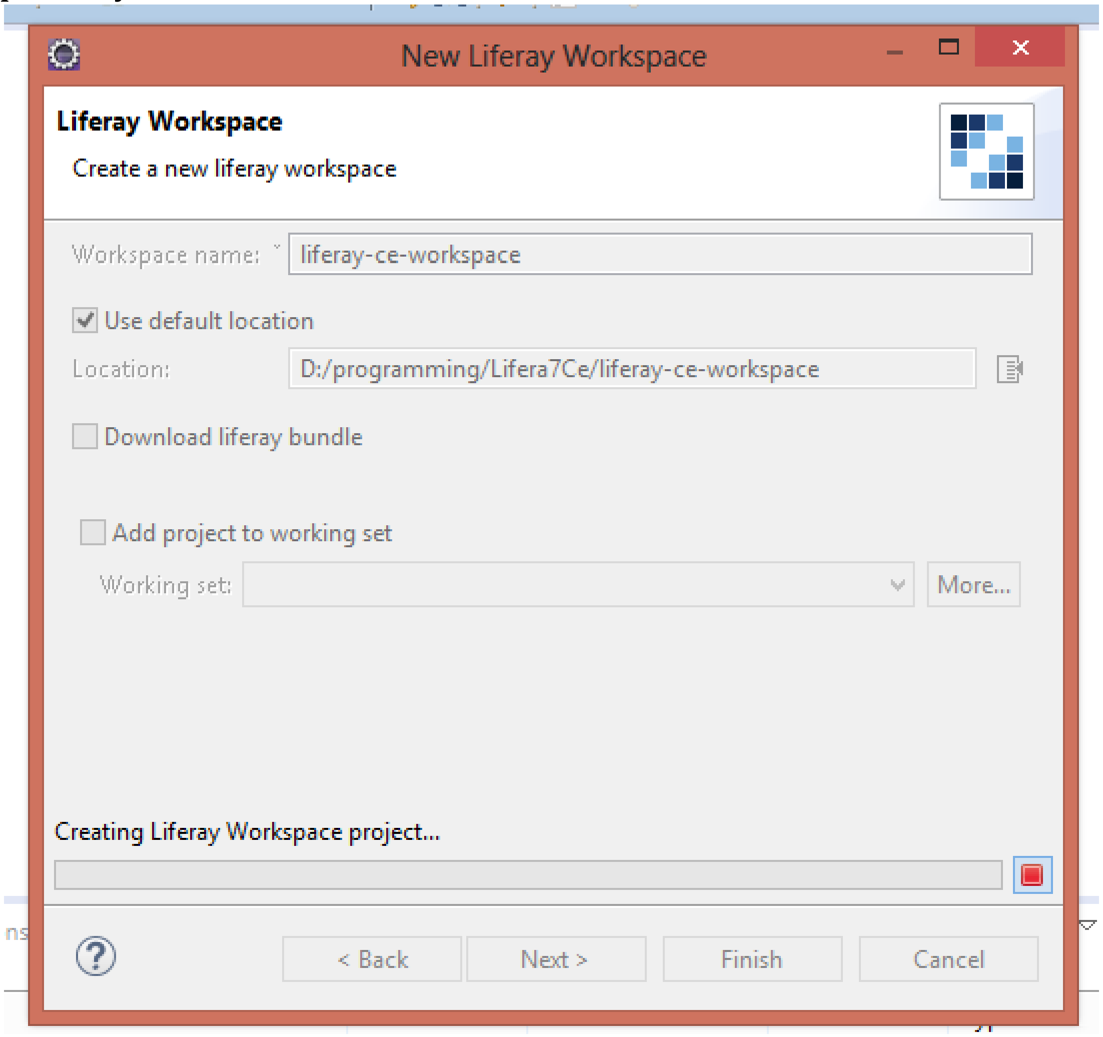
Writable

Loading tasks of all Gra...e projects: (100%)

Step 3: Create Liferay Workspace



In eclipse, click **on File-> New -> Liferay Workspace Project** and provide workspace name. It will download all required libraries from gradle repository.



Liferay Workspace

Create a new liferay workspace

Workspace name: liferay-ce-workspace

☑ Use default location

Location: D:/programming/Lifera7Ce/liferay-ce-workspace

☐ Download liferay bundle

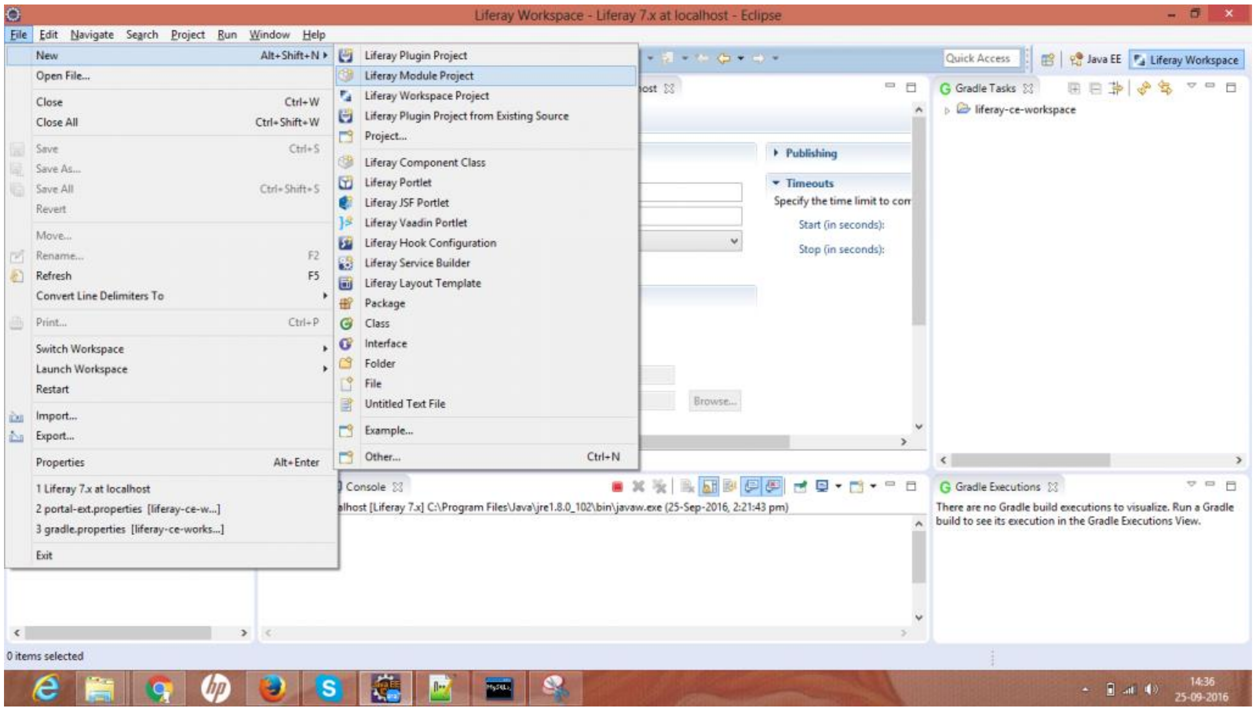
☐ Add project to working set

Working set:

Creating Liferay Workspace project...

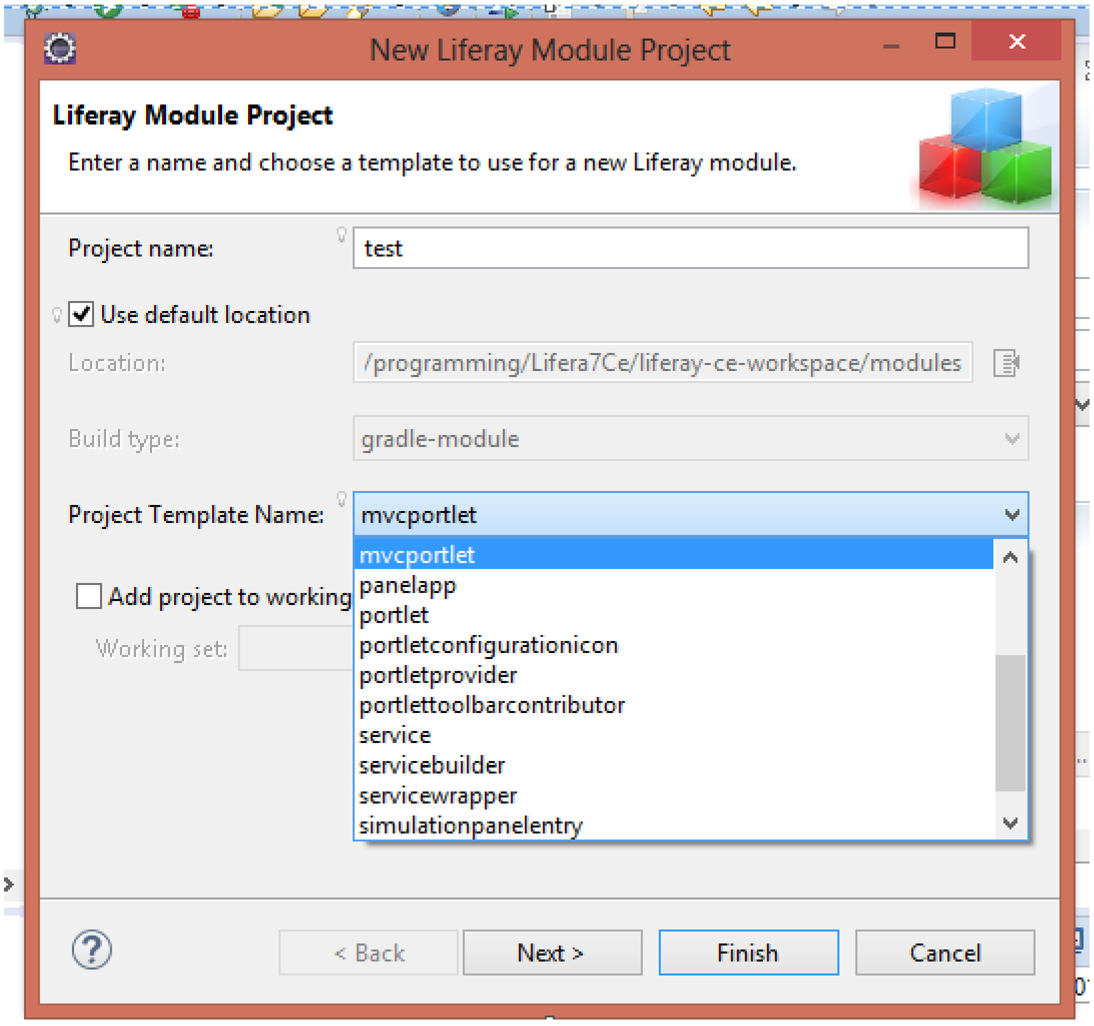
Step 4: Create Liferay Module

Liferay new module wizard: Click **File-> New -> Liferay Module Project**



Select Module configuration:

Liferay7 provides "Project Template Name" to create a modules. You can quicky create service builder, JSP fragment hook, MVC Portlet etc. Here we select module type as "MVC Portlet". Select "**mvcPortlet**" and click on **Finish**.



New Liferay Module Project

Liferay Module Project

Enter a name and choose a template to use for a new Liferay module.

Project name: test

☑ Use default location

Location: /programming/Lifera7Ce/liferay-ce-workspace/modules

Build type: gradle-module

Project Template Name: mvcportlet

- mvcportlet
- panelapp
- portlet
- portletconfigurationicon
- portletprovider
- portlettoolbarcontributor
- service
- servicebuilder
- servicewrapper
- simulationpanelentry

☐ Add project to working

Working set:

•

Step 5 : Liferay 7 Bundles auto configuration

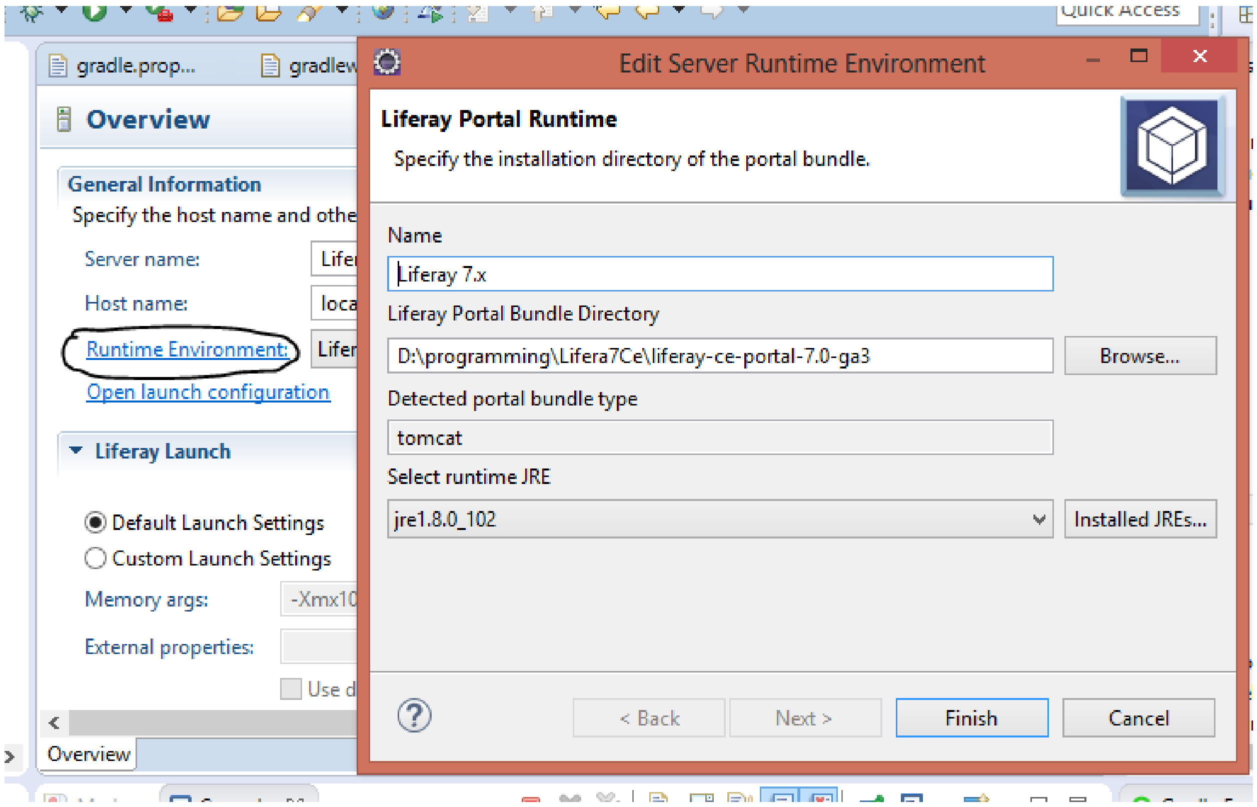
In Liferay 7 community edition, module auto deployment path need to configure and can done in two ways

Liferay tomcat runtime:

Click on Liferay 7x

click on runtime

Update Liferay Portal bundle directory with liferay home directory as shown in below



Edit Server Runtime Environment

Liferay Portal Runtime

Specify the installation directory of the portal bundle.

Name: Liferay 7.x

Liferay Portal Bundle Directory: D:\programming\Lifera7Ce\liferay-ce-portal-7.0-ga3

Detected portal bundle type: tomcat

Select runtime JRE: jre1.8.0_102

gradle.properties

- open gradle.properties file in liferay-workspace folder and add the below property

```
 liferay.workspace.home.dir=D:/programming/Lifera7Ce/liferay-ce-
portal-7.0-ga3
```

Step 6: Liferay 7 Database configuration

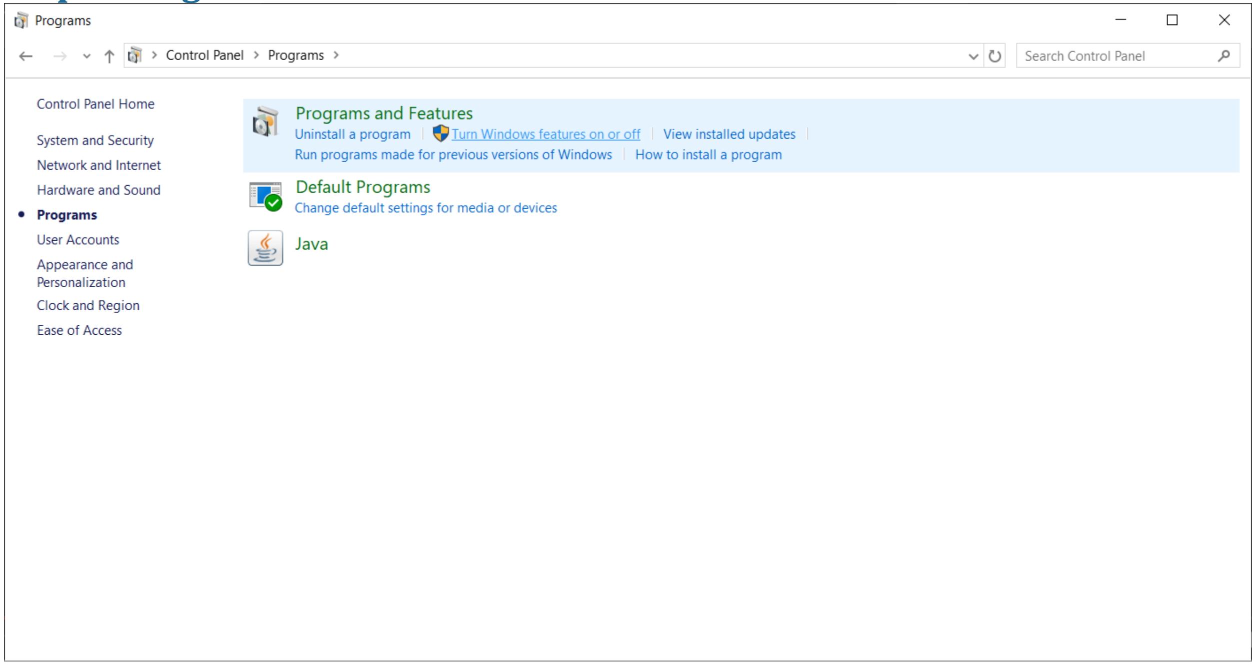
Create database in mysql : create database lportal7_ce character set utf8;
Create **portal-ext.properties** file and add below database properties and liferay home.

```
jdbc.default.url=jdbc\:mysql\://localhost:3307/lportal7_ce?useUnicode\
=true&characterEncoding\=UTF-8&useFastDateParsing\=false

jdbc.default.driverClassName=com.mysql.jdbc.Driver

jdbc.default.username=root

jdbc.default.password=root
```

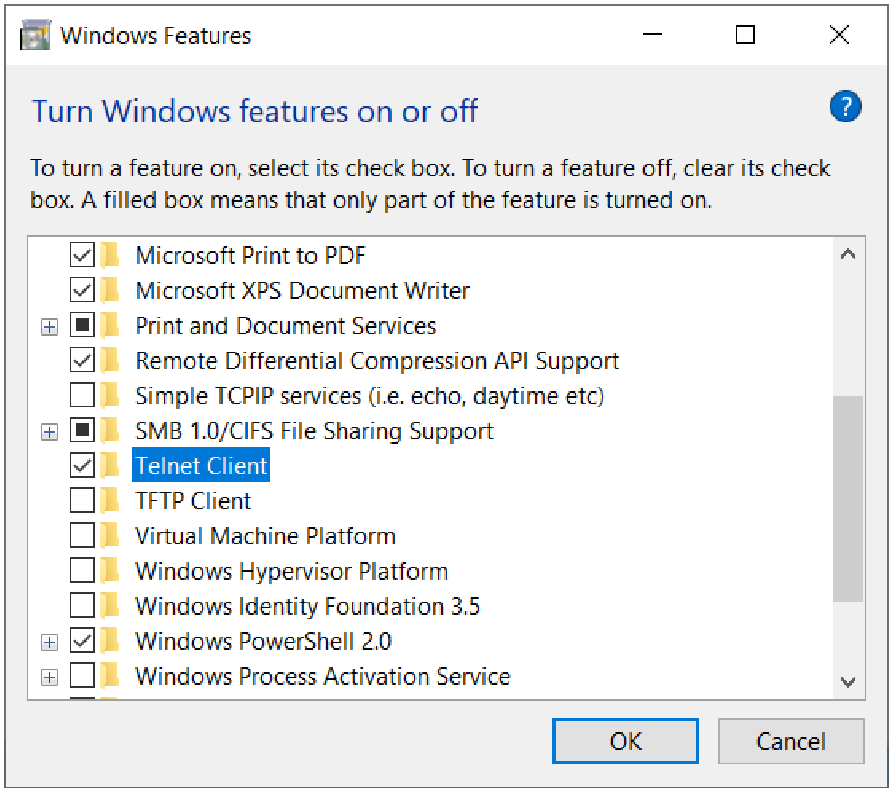
```
liferay.home=D:/programming/Lifera7Ce/liferay-ce-portal-7.0-ga3
```

Copy the mysql to **tomcat\lib\ext** if not is not exist and start the server and you will get config page and complete the setup
If you user oracle

Step 7: Go go shell command



control panel/ turn on telnet



Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ✓ Microsoft Print to PDF
- ✓ Microsoft XPS Document Writer
- ■ Print and Document Services
- ✓ Remote Differential Compression API Support
- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ■ SMB 1.0/CIFS File Sharing Support
- ✓ Telnet Client
- ☐ TFTP Client
- ☐ Virtual Machine Platform
- ☐ Windows Hypervisor Platform
- ☐ Windows Identity Foundation 3.5
- ✓ Windows PowerShell 2.0
- ☐ Windows Process Activation Service

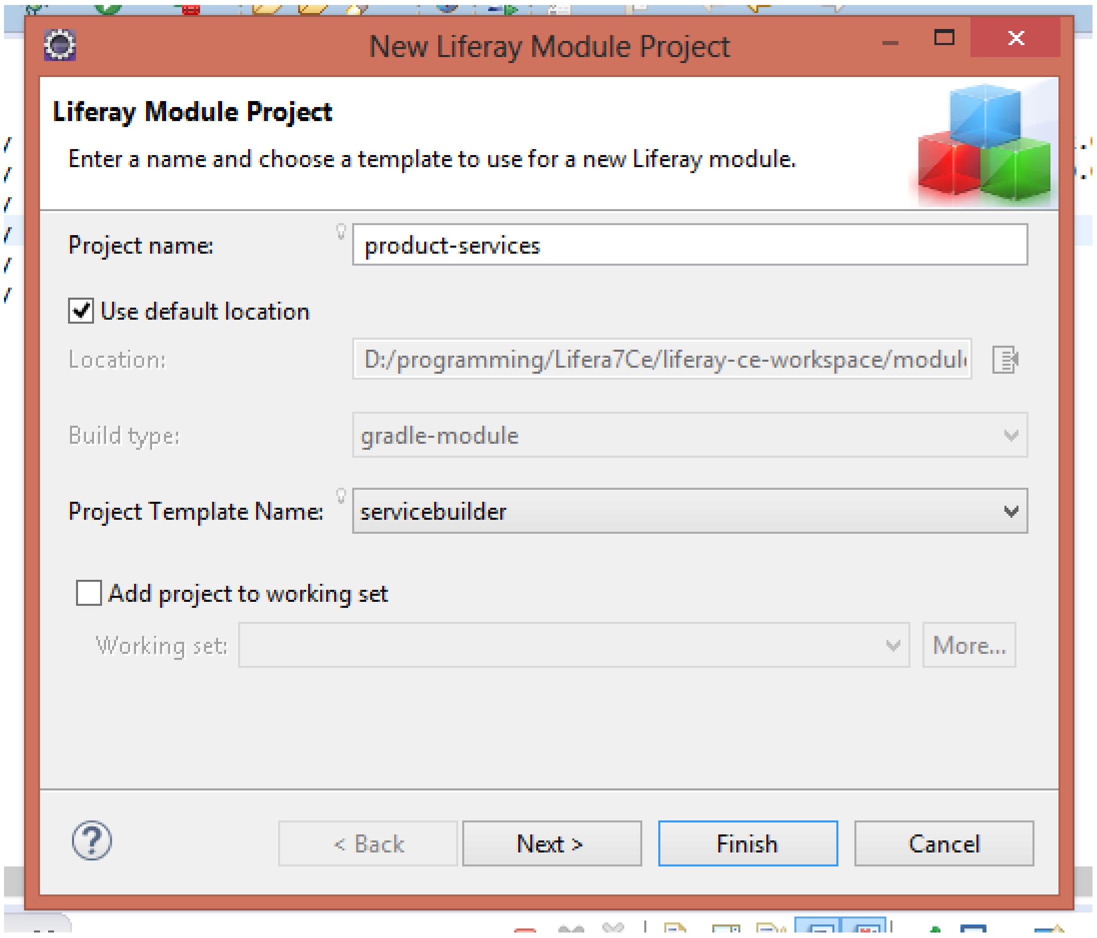
```
telnet localhost 11311
help
lb
stop 32
start 32
```

3. Liferay7 service builder

Create a module of type "Service Builder" and two sub modules are created.

- Api: This module holds all service builder interfaces such as sevice, modal
- Service: This module holds all implementation logic

Create a new service builder module in liferay7: Click on **File->New->Liferay module project**



New Liferay Module Project

Liferay Module Project

Enter a name and choose a template to use for a new Liferay module.

Project name: product-services

[✓] Use default location

Location: D:/programming/Lifera7Ce/liferay-ce-workspace/modul

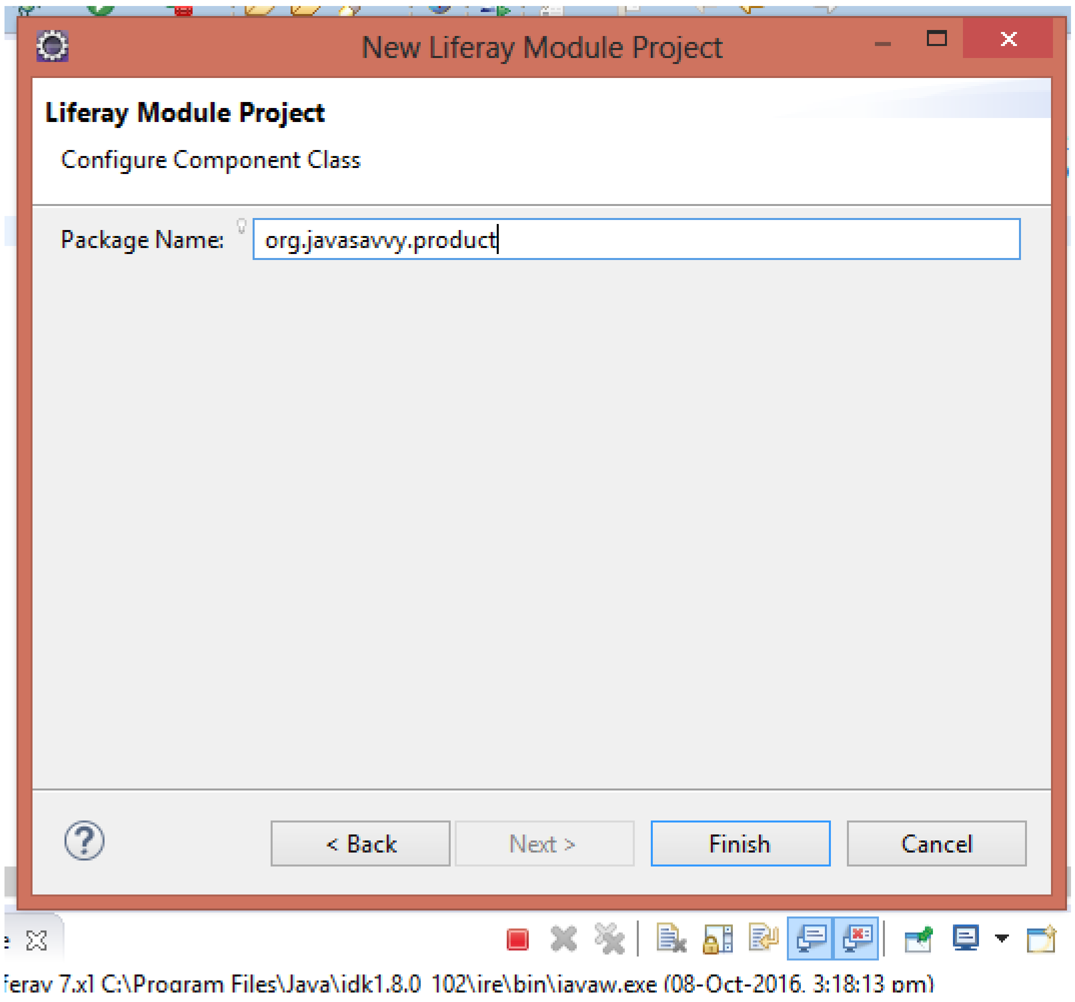
Build type: gradle-module

Project Template Name: servicebuilder

[ ] Add project to working set

Working set:

Select Project Template as “serviceBuilder” and provide package name



New Liferay Module Project

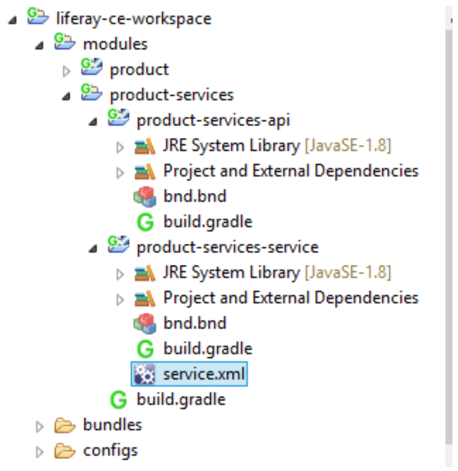
Liferay Module Project

Configure Component Class

Package Name: org.javasavvy.product

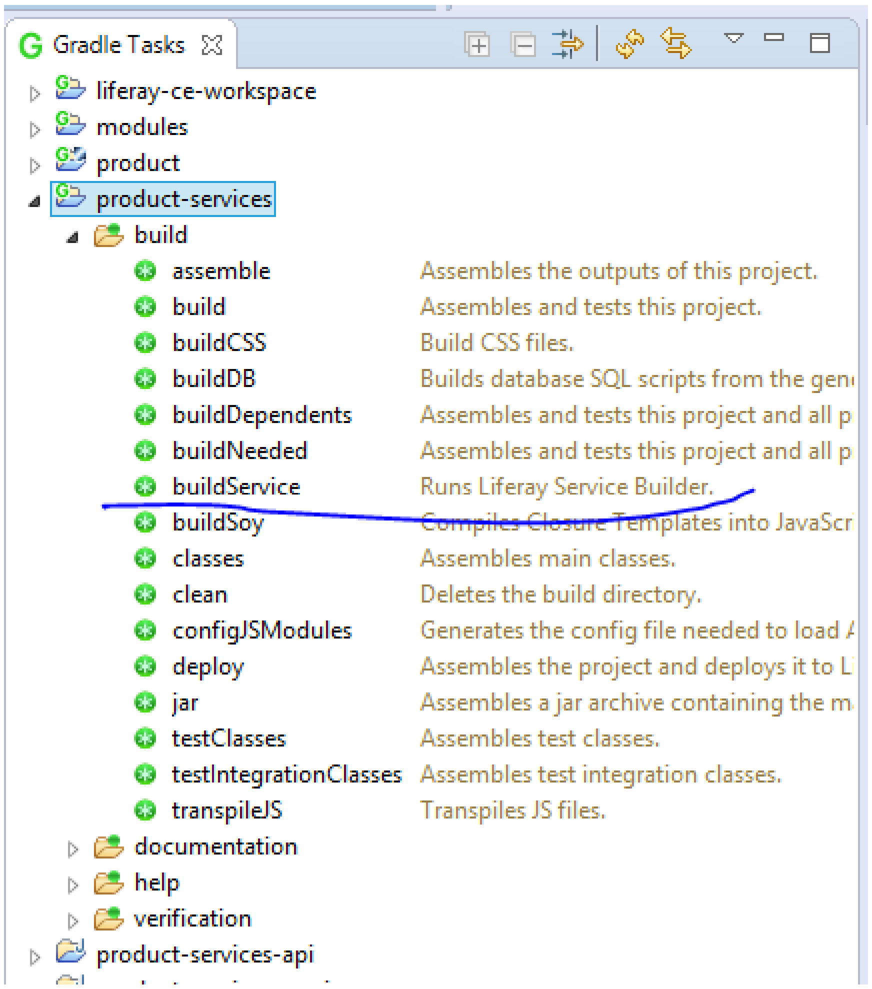
```
ferav 7.x] C:\Program Files\Java\jdk1.8.0_102\jre\bin\javaw.exe (08-Oct-2016, 3:18:13 pm)
```

It will generate two modules as shown in below



```
liferay-ce-workspace
  modules
    product
    product-services
      product-services-api
        JRE System Library [JavaSE-1.8]
        Project and External Dependencies
        bnd.bnd
        build.gradle
      product-services-service
        JRE System Library [JavaSE-1.8]
        Project and External Dependencies
        bnd.bnd
        build.gradle
        service.xml
      build.gradle
  bundles
  configs
```

Liferay Service builder generates module called “product-services”. It again will contain “product-services-api” and “product-services-service” sub modules.
service.xml file also will be created in product-services-service folder with sample entity “Foo”.
Remove the entity **Foo** and add Product
Now run the service builder from Gradle Tasks – **buildService**. You can also run from project explorer : Right click on **Product-Service->Liferay->Gradle->build-service**. It will generate the all service and implementation layer code.


Gradle Tasks

```
▷ liferay-ce-workspace
▷ modules
▷ product
◢ product-services
   ◢ build
        assemble                  Assembles the outputs of this project.
        build                     Assembles and tests this project.
        buildCSS                  Build CSS files.
        buildDB                   Builds database SQL scripts from the gen
        buildDependents           Assembles and tests this project and all p
        buildNeeded               Assembles and tests this project and all p
        buildService              Runs Liferay Service Builder.
        buildSoy                  Compiles Closure Templates into JavaScr
        classes                   Assembles main classes.
        clean                     Deletes the build directory.
        configJSModules           Generates the config file needed to load A
        deploy                    Assembles the project and deploys it to L
        jar                       Assembles a jar archive containing the m
        testClasses               Assembles test classes.
        testIntegrationClasses    Assembles test integration classes.
        transpileJS               Transpiles JS files.
   ▷ documentation
   ▷ help
   ▷ verification
▷ product-services-api
```

product-services-api contains service interfaces and wrapper services like in Liferay6

- contains modal interfaces – Product.java

- service layer interface and LocalServiceUtil.java
- persistence layer interface classes and Utilities classes
- common exception classes

Now Service builder is generated and how to add service builder dependency in another module?

Now our UI code is in product module that requires "product-services" as dependency

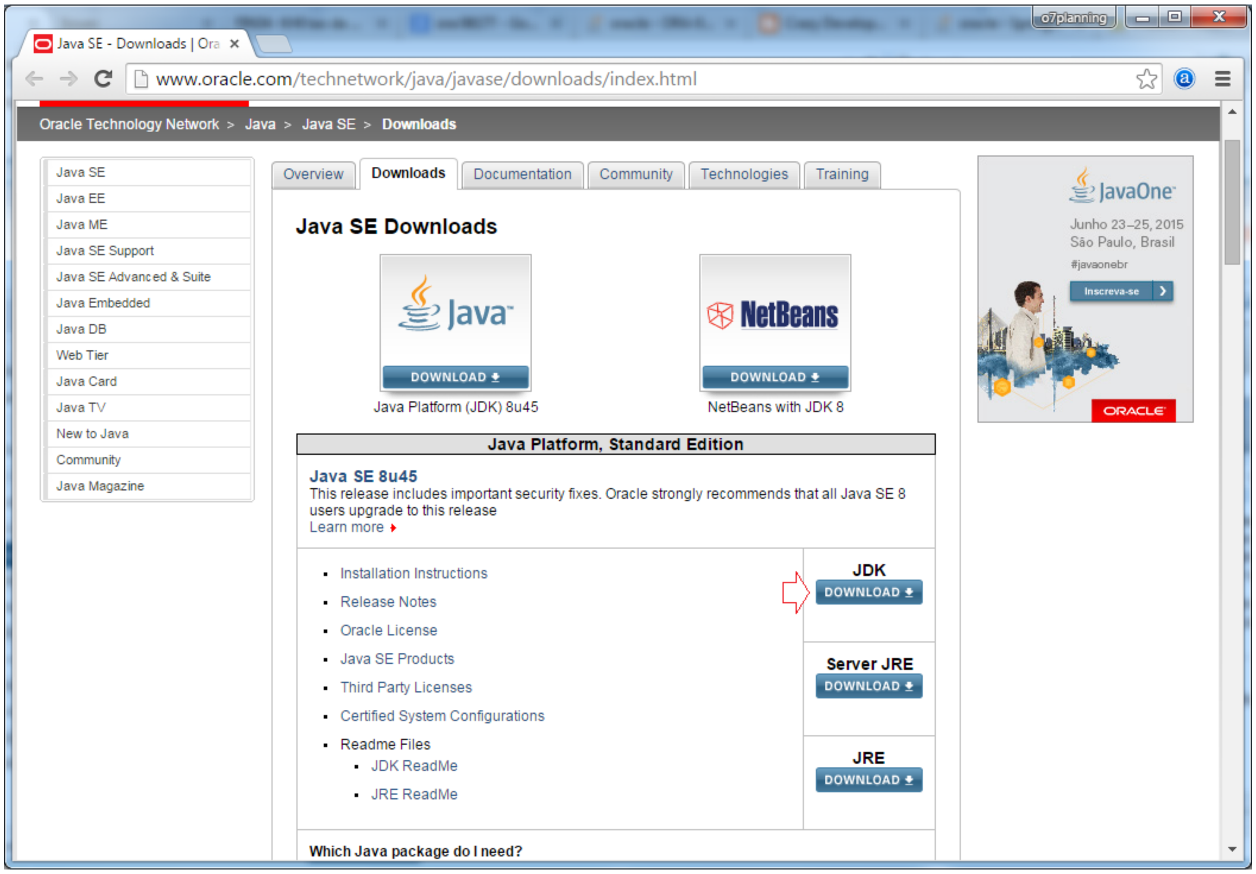
open the build.gradle file in product module

add below gradle depency

```
dependencies {
 compileOnly group: "com.liferay.portal", name:
"com.liferay.portal.kernel", version: "2.0.0"
 compileOnly group: "com.liferay.portal", name:
"com.liferay.util.taglib", version: "2.0.0"
 compileOnly group: "javax.portlet", name: "portlet-api", version:
"2.0"
 compileOnly group: "javax.servlet", name: "servlet-api", version:
"2.5"
 compileOnly group: "jstl", name: "jstl", version: "1.2"
 compileOnly group: "org.osgi", name: "org.osgi.compendium", version:
"5.0.0"
// ADD
 compileOnly project(":modules:product-services:product-services-api")
}
```

4. Download JDK

- http://www.oracle.com/technetwork/java/javase/downloads/index.html



Java SE Downloads

Java — DOWNLOAD — Java Platform (JDK) 8u45

NetBeans — DOWNLOAD — NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u45
This release includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release
Learn more

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
  - JDK ReadMe
  - JRE ReadMe

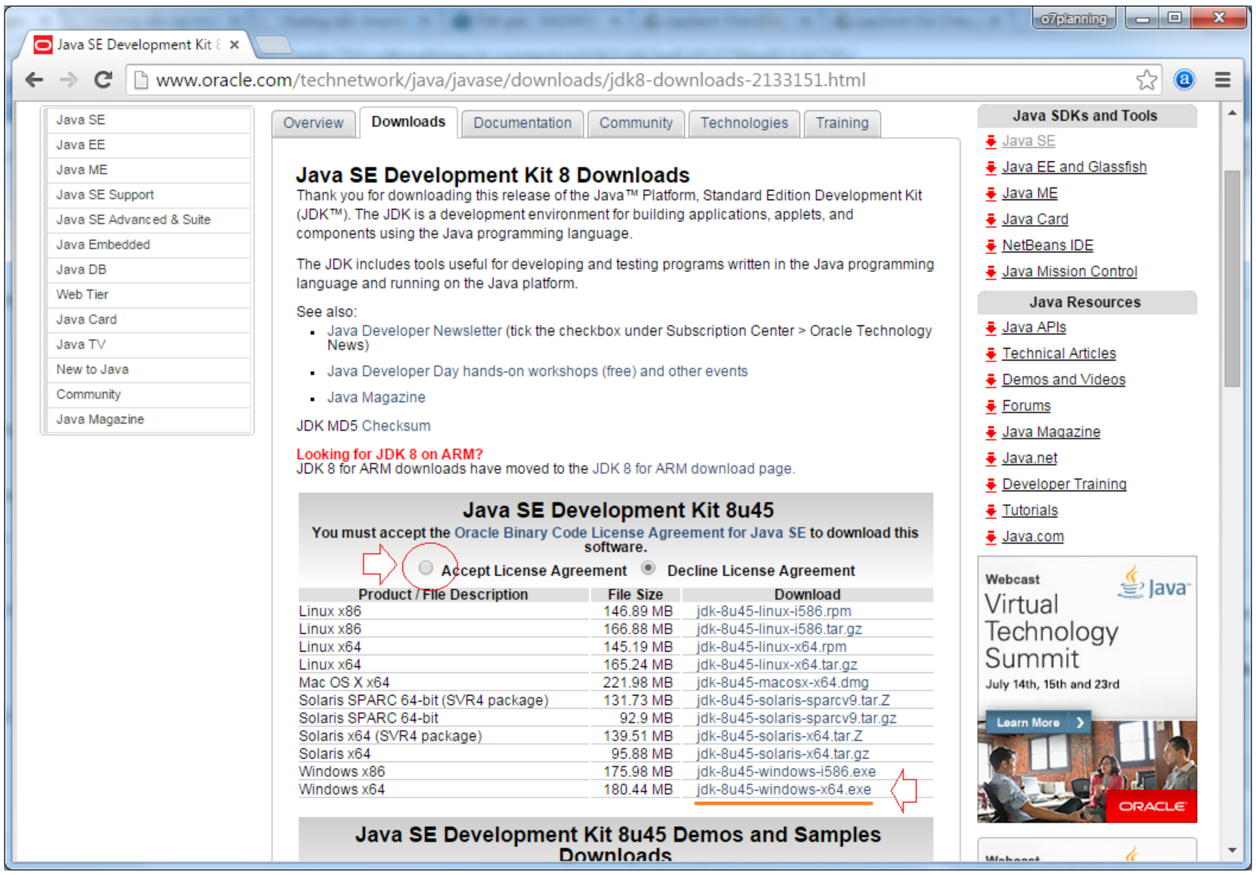
JDK DOWNLOAD

Server JRE DOWNLOAD

JRE DOWNLOAD

Which Java package do I need?

Bạn nên kiểm tra xem hệ điều hành của mình là 32bit hay 64bit để download bộ cài **JDK** phù hợp.



Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

Looking for JDK 8 on ARM?
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

Java SE Development Kit 8u45

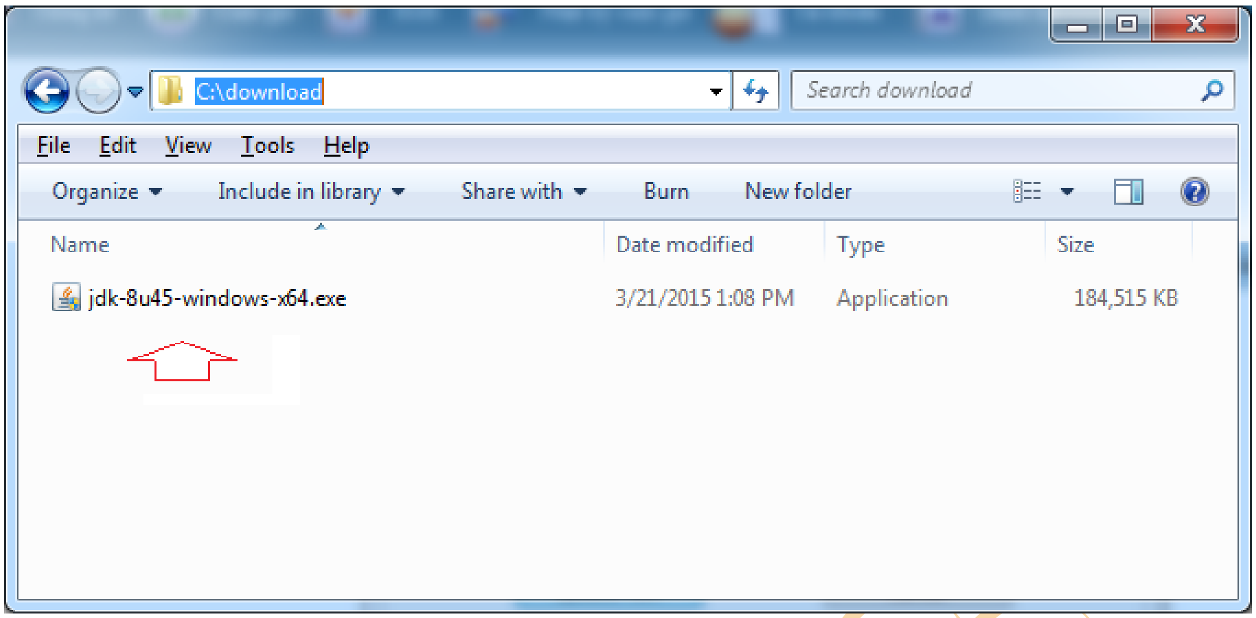
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

( ) Accept License Agreement (●) Decline License Agreement

Product / File Description	File Size	Download
Linux x86	146.89 MB	jdk-8u45-linux-i586.rpm
Linux x86	166.88 MB	jdk-8u45-linux-i586.tar.gz
Linux x64	145.19 MB	jdk-8u45-linux-x64.rpm
Linux x64	165.24 MB	jdk-8u45-linux-x64.tar.gz
Mac OS X x64	221.98 MB	jdk-8u45-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	131.73 MB	jdk-8u45-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	92.9 MB	jdk-8u45-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	139.51 MB	jdk-8u45-solaris-x64.tar.Z
Solaris x64	95.88 MB	jdk-8u45-solaris-x64.tar.gz
Windows x86	175.98 MB	jdk-8u45-windows-i586.exe
Windows x64	180.44 MB	jdk-8u45-windows-x64.exe

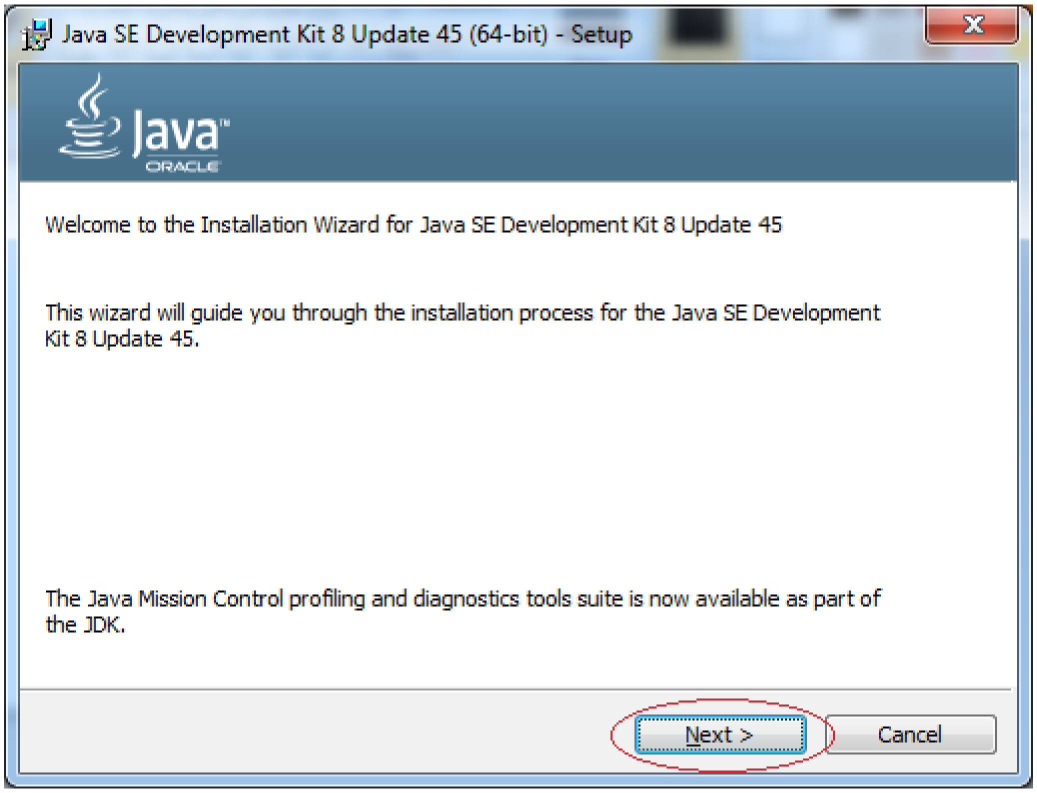
Java SE Development Kit 8u45 Demos and Samples Downloads

Kết quả download được:



Name	Date modified	Type	Size
jdk-8u45-windows-x64.exe	3/21/2015 1:08 PM	Application	184,515 KB

Cài đặt Java


Java SE Development Kit 8 Update 45 (64-bit) - Setup

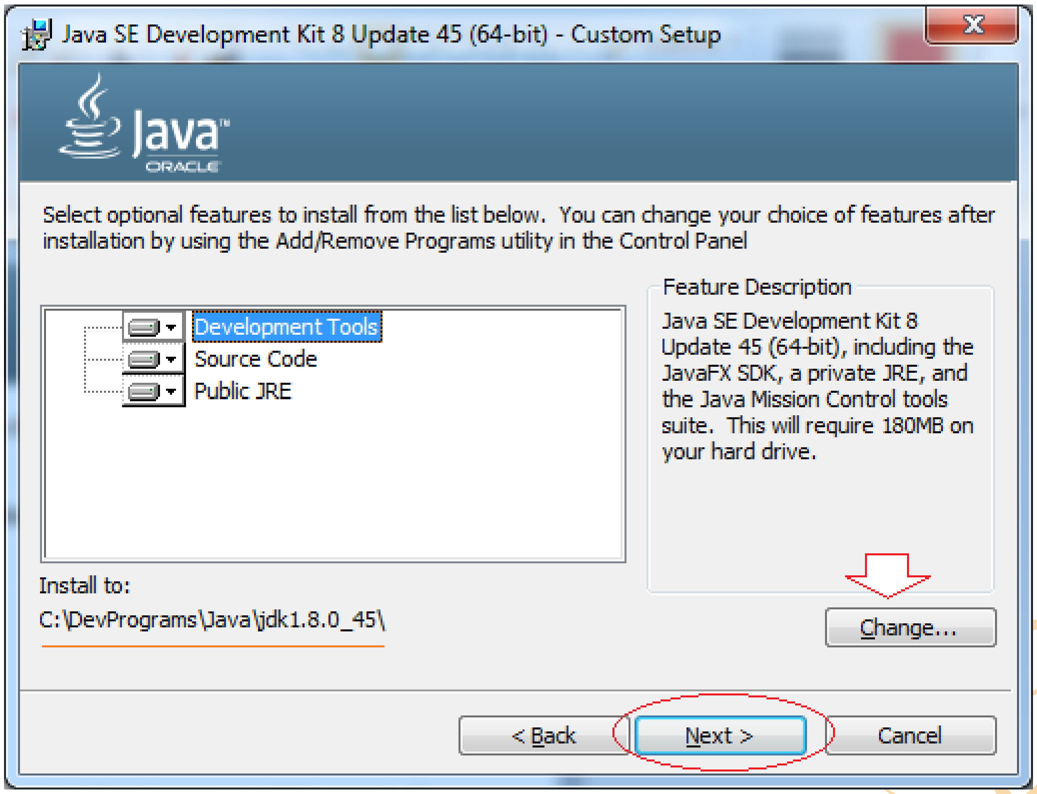
Welcome to the Installation Wizard for Java SE Development Kit 8 Update 45

This wizard will guide you through the installation process for the Java SE Development Kit 8 Update 45.

The Java Mission Control profiling and diagnostics tools suite is now available as part of the JDK.

Nhập vào thư mục mà JDK sẽ được cài đặt ra, ở đây tôi đặt là:

- **C:\DevPrograms\Java\jdk1.8.0_45\**


Java SE Development Kit 8 Update 45 (64-bit) - Custom Setup

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

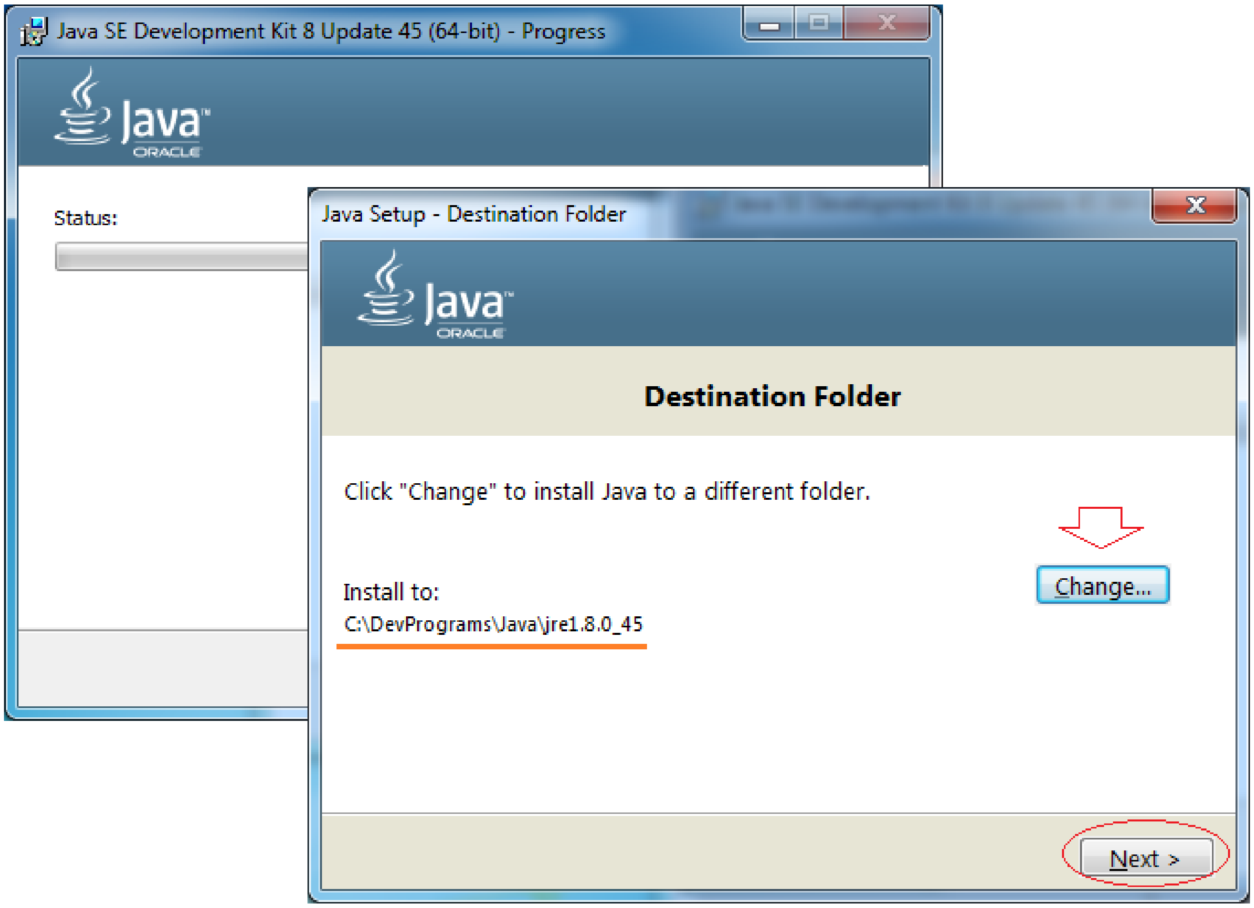
Development Tools
Source Code
Public JRE

Feature Description
Java SE Development Kit 8 Update 45 (64-bit), including the JavaFX SDK, a private JRE, and the Java Mission Control tools suite. This will require 180MB on your hard drive.

Install to:
C:\DevPrograms\Java\jdk1.8.0_45\

Ngay sau khi cài đặt xong JDK, bộ cài đặt sẽ tiếp tục hỏi bạn vị trí JRE sẽ được cài ra. Ở đây tôi chọn:

- **C:\DevPrograms\Java\jre.8.0_45\**

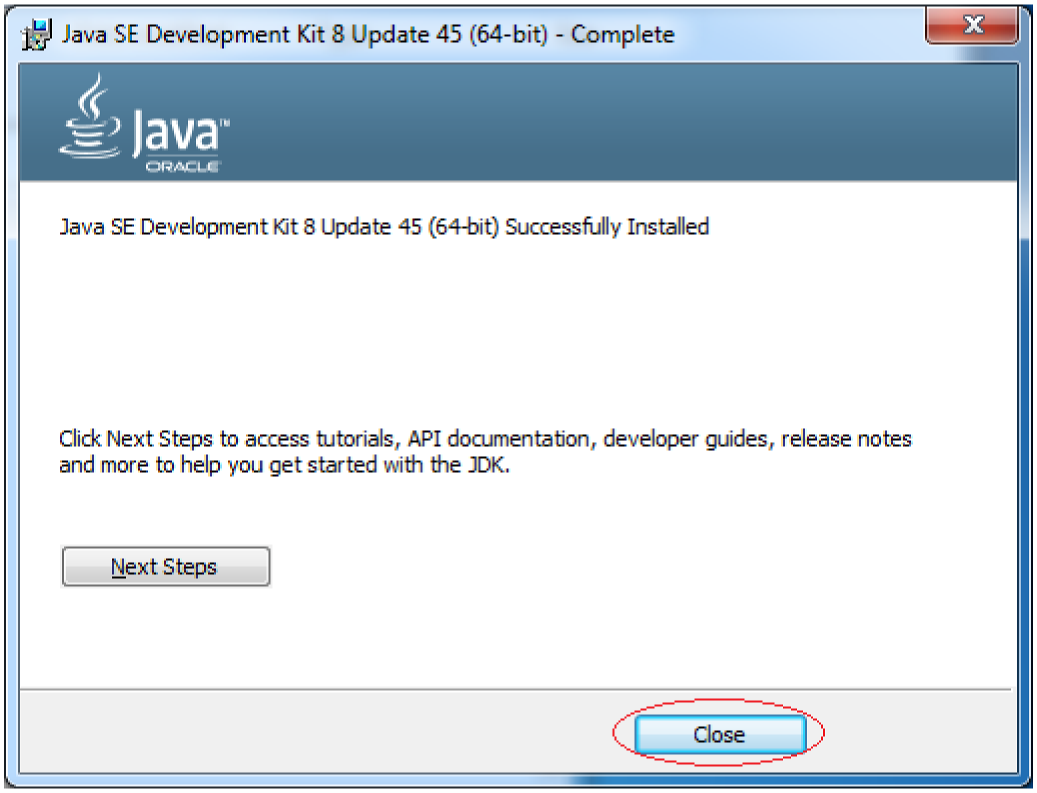

Java SE Development Kit 8 Update 45 (64-bit) - Progress
Status:
Java Setup - Destination Folder
Java
ORACLE
Change...
Next >

Destination Folder

Click "Change" to install Java to a different folder.

Install to:
C:\DevPrograms\Java\jre1.8.0_45

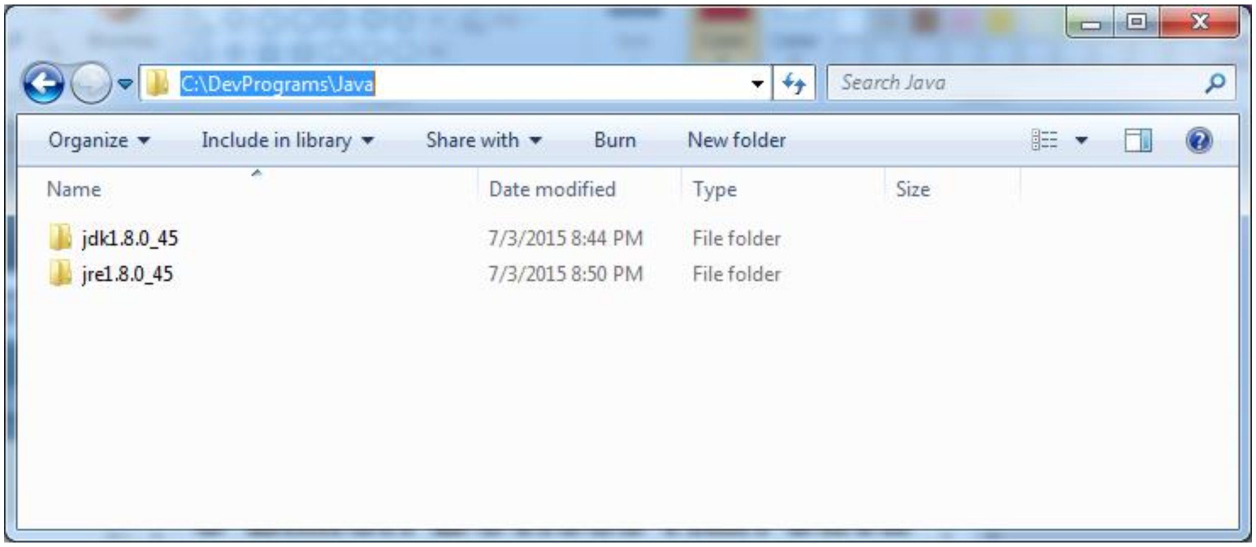
Java đã được cài đặt thành công.


Java SE Development Kit 8 Update 45 (64-bit) - Complete

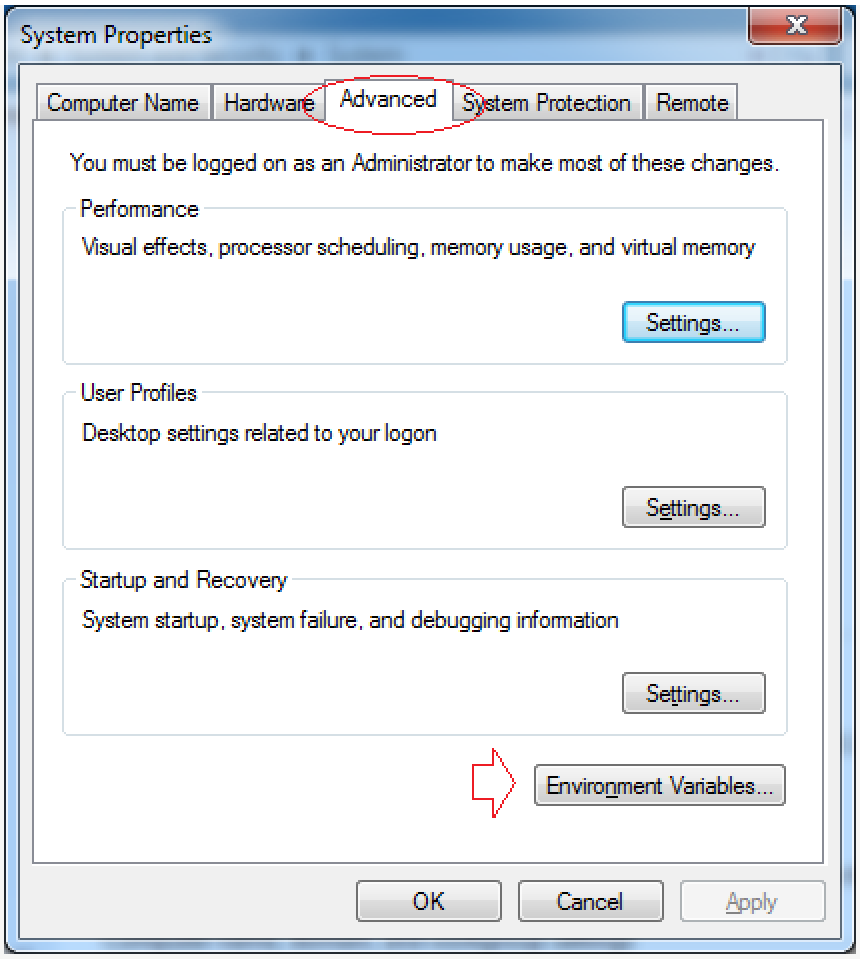
Java SE Development Kit 8 Update 45 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

Kết quả bạn có được 2 thư mục:


C:\DevPrograms\Java

Name	Date modified	Type	Size
jdk1.8.0_45	7/3/2015 8:44 PM	File folder	
jre1.8.0_45	7/3/2015 8:50 PM	File folder	


System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

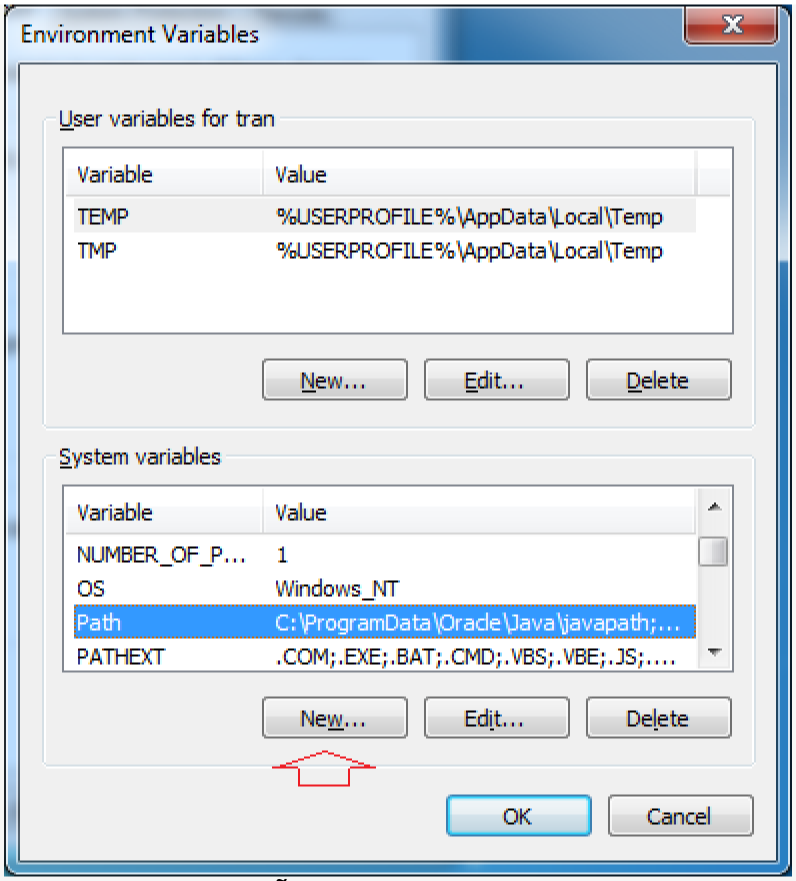
User Profiles

Desktop settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

Nhấn ***New*** để tạo mới một biến môi trường có tên **JAVA_HOME**.

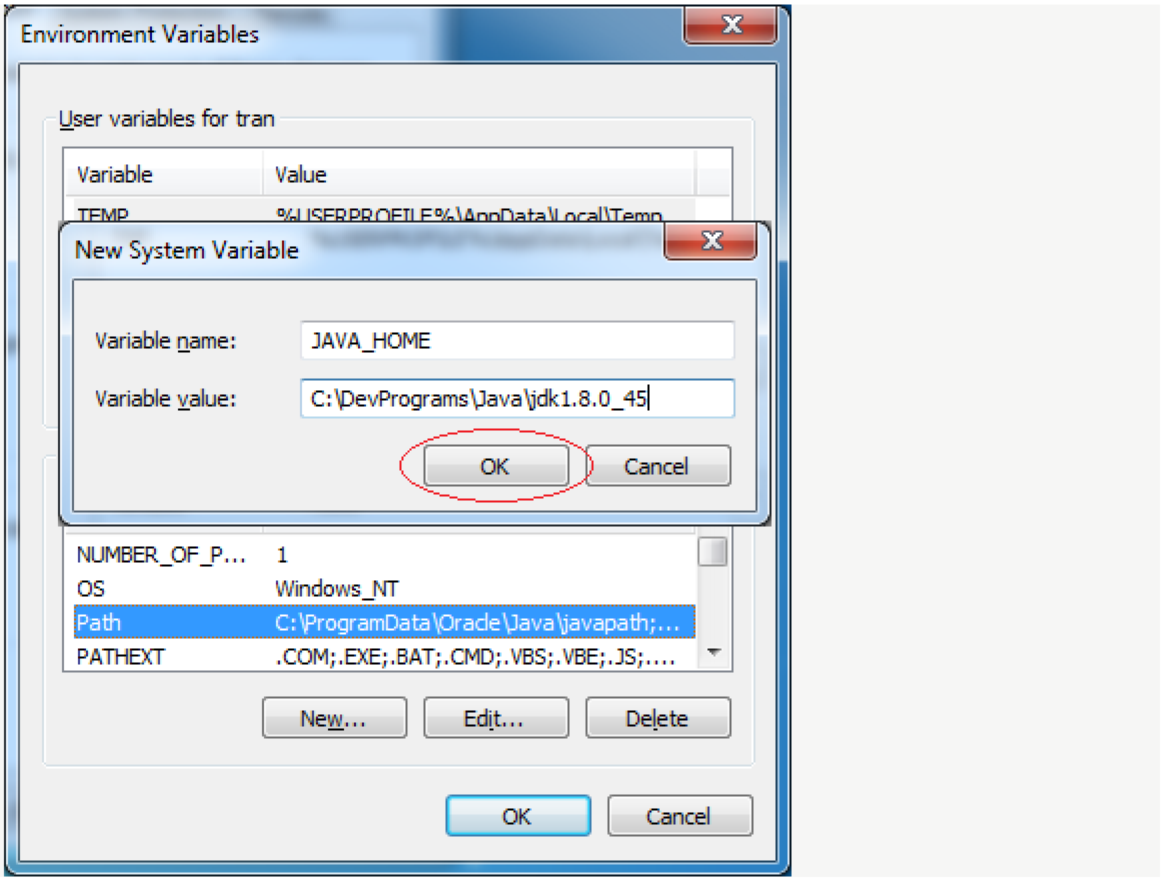


Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
NUMBER_OF_P...	1
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Nhập vào đường dẫn tới thư mục JDK.

- **Variable name:** JAVA_HOME
- **Variable value:** C:\DevPrograms\Java\jdk1.8.0_45


Environment Variables

User variables for tran

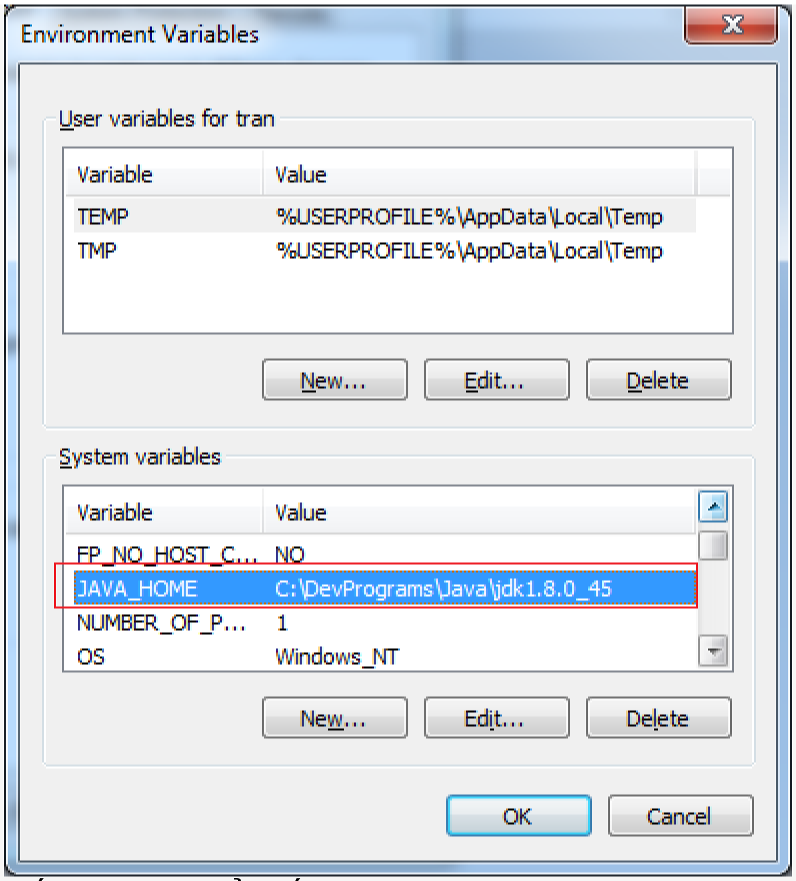
Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp

New System Variable

Variable name: JAVA_HOME

Variable value: C:\DevPrograms\Java\jdk1.8.0_45

Variable	Value
NUMBER_OF_P...	1
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....



Environment Variables

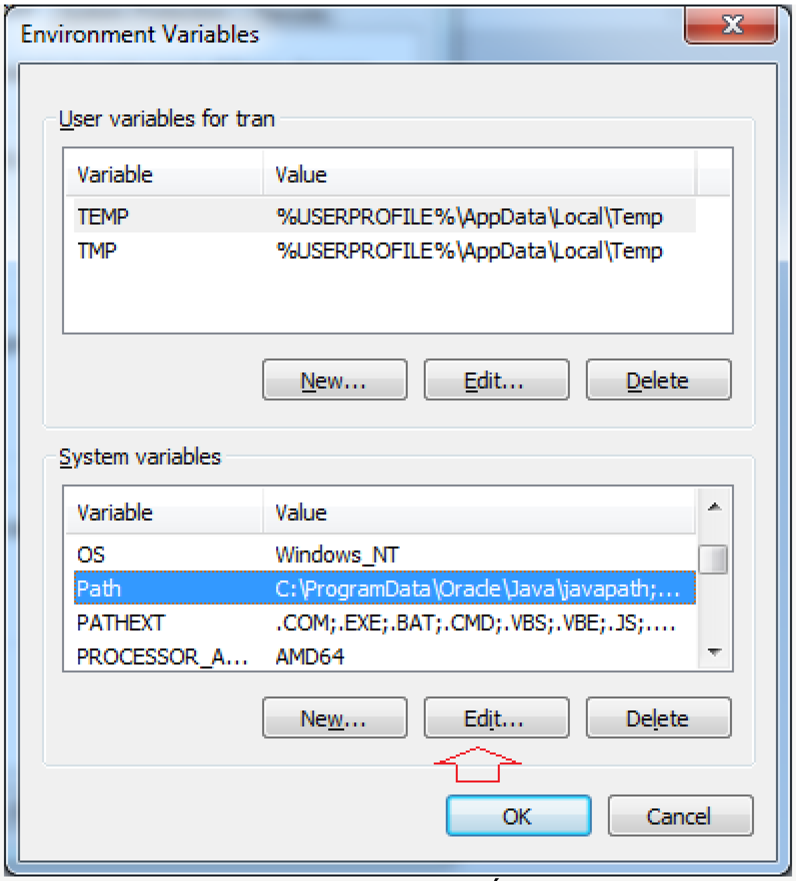
User variables for tran

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
FP_NO_HOST_C...	NO
JAVA_HOME	C:\DevPrograms\Java\jdk1.8.0_45
NUMBER_OF_P...	1
OS	Windows_NT

Tiếp theo sửa đổi biến môi trường ***path***



Environment Variables

User variables for tran

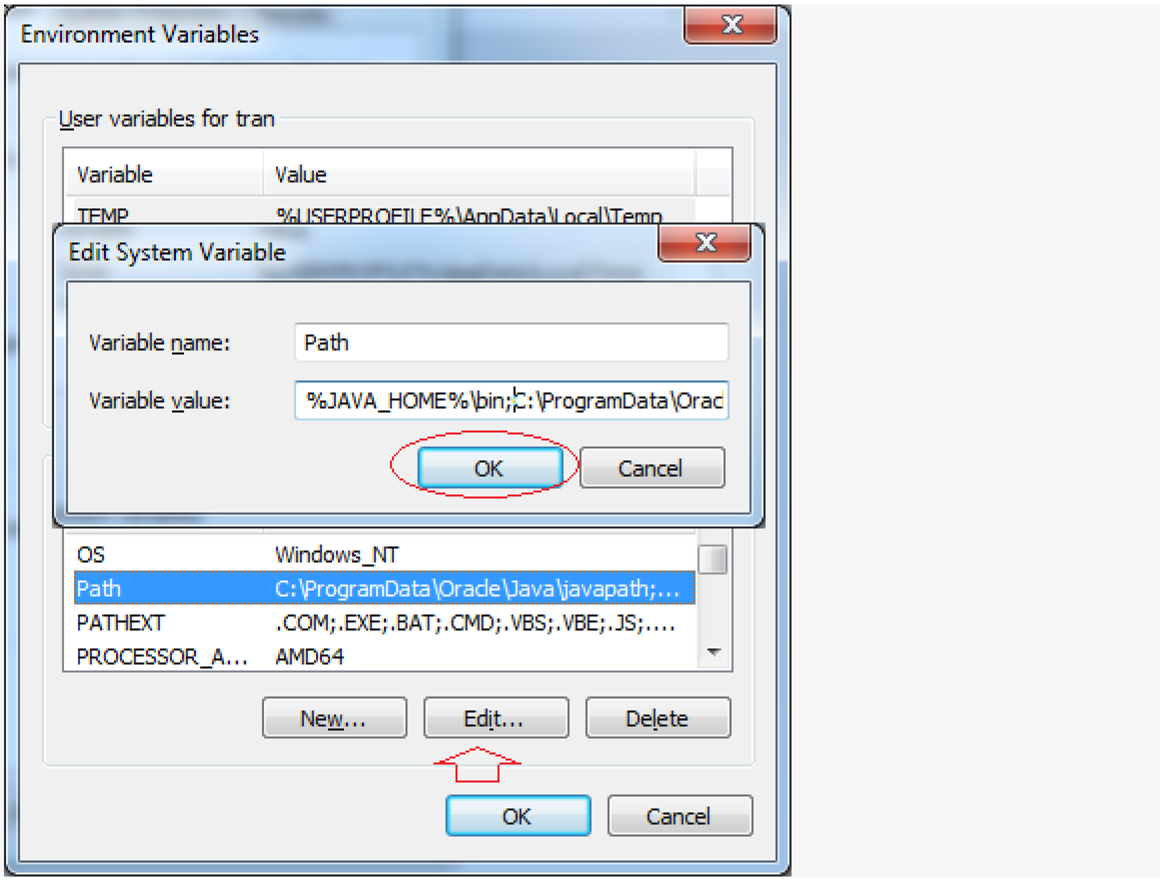
Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

Thêm vào phía trước giá trị của biến môi trường ***path***:

- **%JAVA_HOME%\bin;**



Environment Variables

User variables for tran

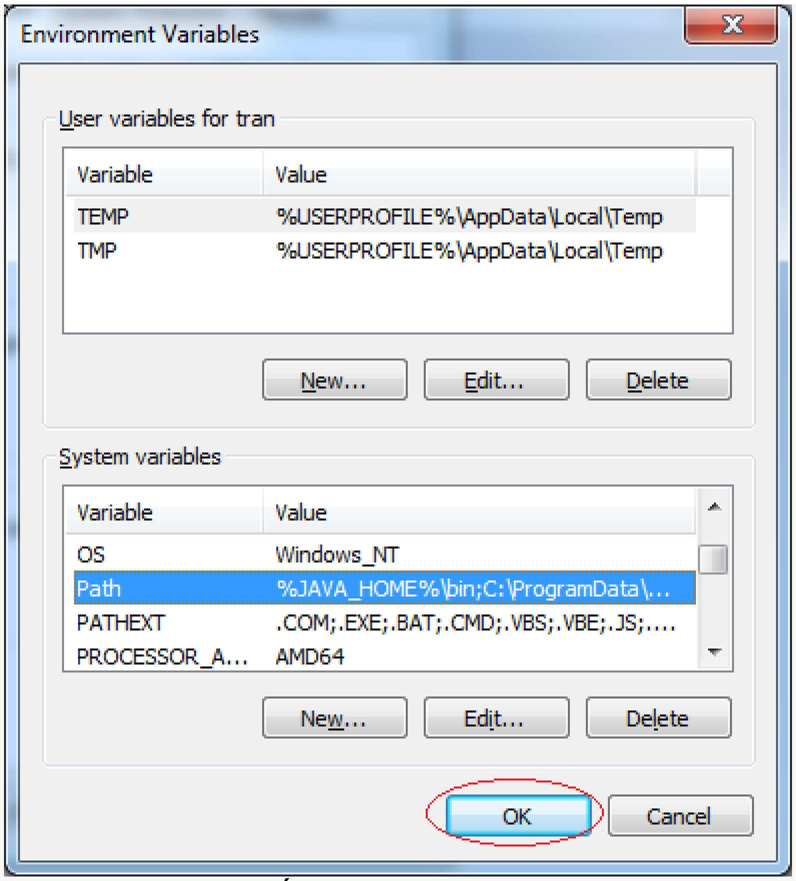
Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp

Edit System Variable

Variable name: Path

Variable value: %JAVA_HOME%\bin;C:\ProgramData\Orac

Variable	Value
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64



Environment Variables

User variables for tran

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
OS	Windows_NT
Path	%JAVA_HOME%\bin;C:\ProgramData\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

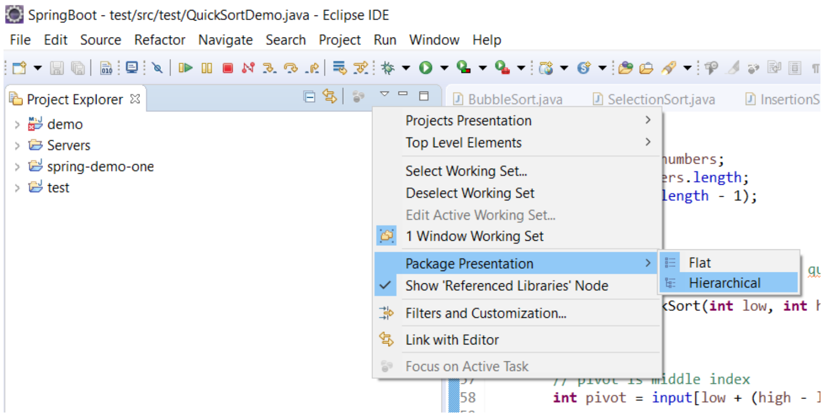
New... Edit... Delete

OK Cancel

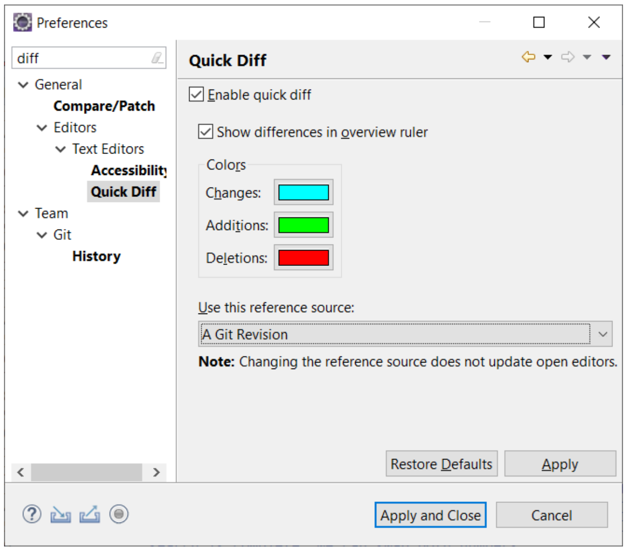
Bạn đã cài đặt và cấu hình Java thành công.

Check: java -version

Cách expand xem list file:



Quick diff



Preferences

diff

- General
  - Compare/Patch
  - Editors
    - Text Editors
      - Accessibility
      - Quick Diff
- Team
  - Git
    - History

Quick Diff

- [✓] Enable quick diff
- [✓] Show differences in overview ruler

Colors
- Changes: 🟦
- Additions: 🟩
- Deletions: 🟥

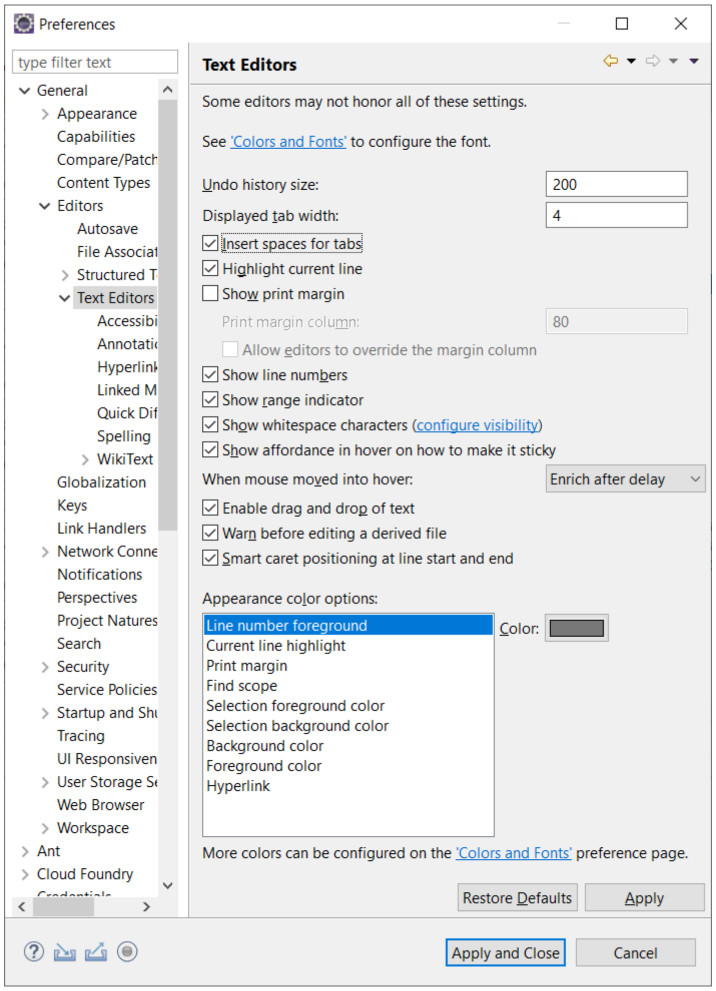
Use this reference source:

A Git Revision

Note: Changing the reference source does not update open editors.

Restore Defaults | Apply

Apply and Close | Cancel



Preferences

type filter text

- General
  - Appearance
  - Capabilities
  - Compare/Patch
  - Content Types
  - Editors
    - Autosave
    - File Associat
    - Structured T
    - Text Editors
      - Accessibi
      - Annotati
      - Hyperlink
      - Linked M
      - Quick Dif
      - Spelling
      - WikiText
  - Globalization
  - Keys
  - Link Handlers
  - Network Conne
  - Notifications
  - Perspectives
  - Project Natures
  - Search
  - Security
  - Service Policies
  - Startup and Shu
  - Tracing
  - UI Responsiven
  - User Storage Se
  - Web Browser
  - Workspace
- Ant
- Cloud Foundry

Text Editors

Some editors may not honor all of these settings.

See 'Colors and Fonts' to configure the font.

Undo history size: 200

Displayed tab width: 4

- [✓] Insert spaces for tabs
- [✓] Highlight current line
- [ ] Show print margin

Print margin column: 80

- [ ] Allow editors to override the margin column
- [✓] Show line numbers
- [✓] Show range indicator
- [✓] Show whitespace characters (configure visibility)
- [✓] Show affordance in hover on how to make it sticky

When mouse moved into hover: Enrich after delay

- [✓] Enable drag and drop of text
- [✓] Warn before editing a derived file
- [✓] Smart caret positioning at line start and end

Appearance color options:

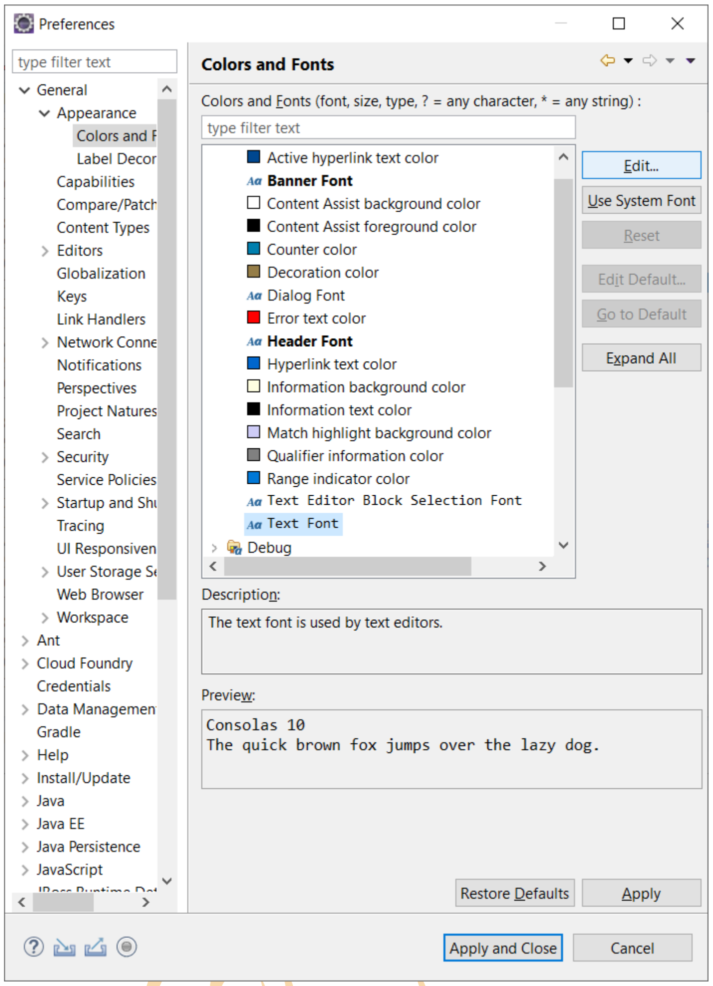
- Line number foreground
- Current line highlight
- Print margin
- Find scope
- Selection foreground color
- Selection background color
- Background color
- Foreground color
- Hyperlink

Color: ■

More colors can be configured on the 'Colors and Fonts' preference page.

Restore Defaults | Apply

Apply and Close | Cancel



Preferences

type filter text

- General
  - Appearance
    - Colors and F
    - Label Decor
  - Capabilities
  - Compare/Patch
  - Content Types
  - Editors
  - Globalization
  - Keys
  - Link Handlers
  - Network Conne
  - Notifications
  - Perspectives
  - Project Natures
  - Search
  - Security
  - Service Policies
  - Startup and Shu
  - Tracing
  - UI Responsiven
  - User Storage Se
  - Web Browser
  - Workspace
- Ant
- Cloud Foundry
- Credentials
- Data Managemen
- Gradle
- Help
- Install/Update
- Java
- Java EE
- Java Persistence
- JavaScript

Colors and Fonts

Colors and Fonts (font, size, type, ? = any character, * = any string) :

type filter text

- Active hyperlink text color
- **Banner Font**
- Content Assist background color
- Content Assist foreground color
- Counter color
- Decoration color
- Dialog Font
- Error text color
- **Header Font**
- Hyperlink text color
- Information background color
- Information text color
- Match highlight background color
- Qualifier information color
- Range indicator color
- `Text Editor Block Selection Font`
- `Text Font`
- Debug

Edit... | Use System Font | Reset | Edit Default... | Go to Default | Expand All

Description:

The text font is used by text editors.

Preview:

```
Consolas 10
The quick brown fox jumps over the lazy dog.
```

Restore Defaults | Apply

Apply and Close | Cancel

5. More config

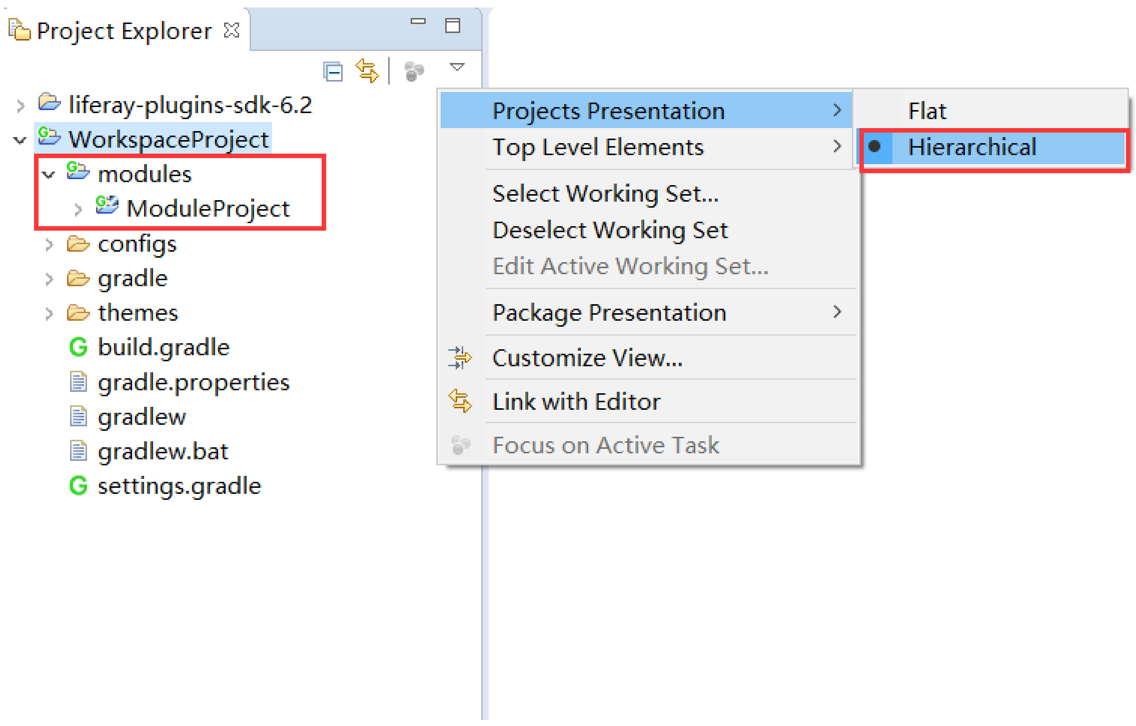
Install a new software

https://portal.liferay.dev/docs/7-0/tutorials/-/knowledge_base/t/installing-liferay-ide

1. In your browser, go to the Liferay IDE page. Copy the URL to the update site you're interested in (stable or milestone).
2. Select *Help* → *Install New Software*.
3. In the *Add* dialog, click the *Archive* button and browse to the location of the downloaded Liferay IDE Zip file.
4. You 'll see the Liferay IDE components in the list below. Check them off and click Next.
5. Accept the terms of the agreements and click Next, and Liferay IDE is installed. Like other Eclipse plugins you ' ll have to restart Eclipse to enable it.

View flat

You can configure your workspace's module presentation by switching between the default *Hierarchical* or *Flat* views. To do this, navigate to the Project Explorer's *View Menu* (▽), select *Projects Presentation* and then select the presentation mode you'd like to display. The Hierarchical view displays subfolders and subprojects under the workspace project, whereas the Flat view displays the workspace's modules separately from the workspace.



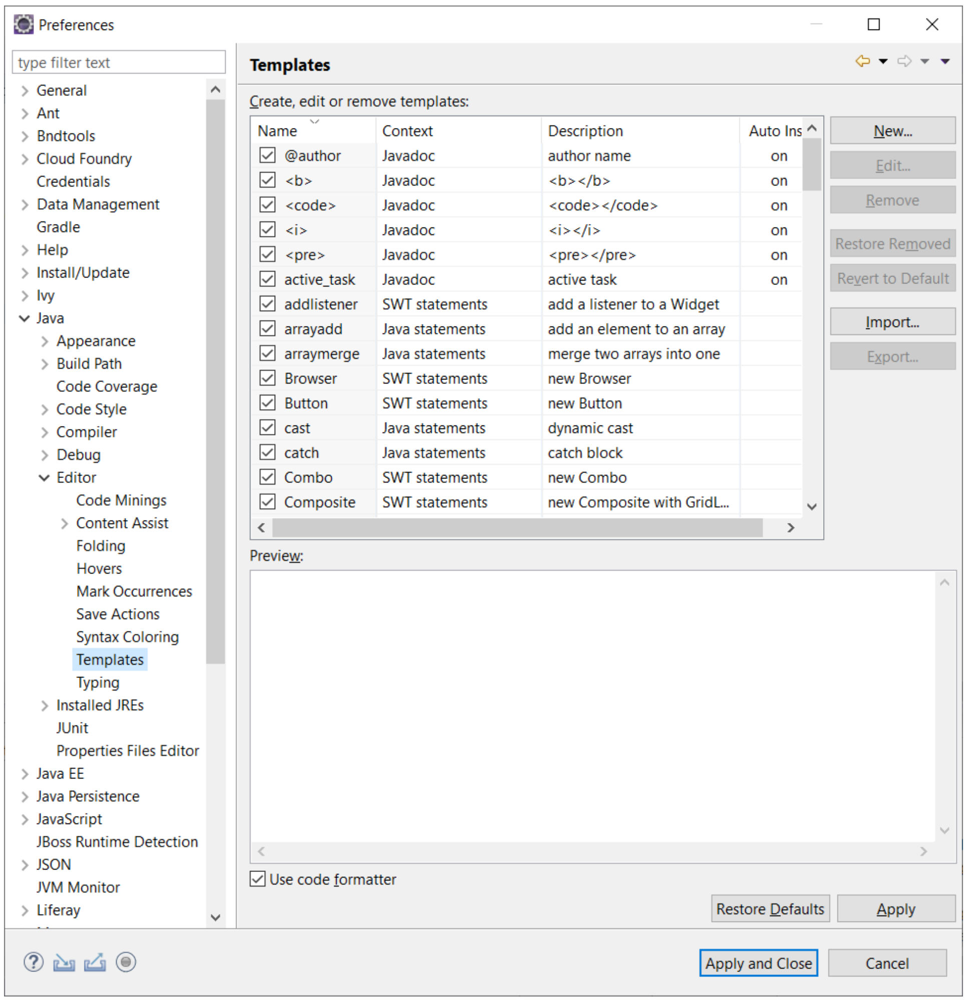
Import template format

https://help.eclipse.org/2019-12/index.jsp?topic=%2Forg.eclipse.cdt.doc.user%2Ftasks%2Fcdt_t_imp_code_temp.htm

To import a template

1. Click Window > Preferences.
2. Expand JAVA, expand Editor and click Templates.
3. Click Import.
4. Select the template file that you want to import.
5. Click OK.

The template list is updated to include the template that you imported.



General
Ant
Bndtools
Cloud Foundry
Credentials
Data Management
Gradle
Help
Install/Update
Ivy
Java
- Appearance
- Build Path
- Code Coverage
- Code Style
- Compiler
- Debug
- Editor
  - Code Minings
  - Content Assist
  - Folding
  - Hovers
  - Mark Occurrences
  - Save Actions
  - Syntax Coloring
  - Templates
  - Typing
- Installed JREs
- JUnit
- Properties Files Editor

Java EE
Java Persistence
JavaScript
JBoss Runtime Detection
JSON
JVM Monitor
Liferay

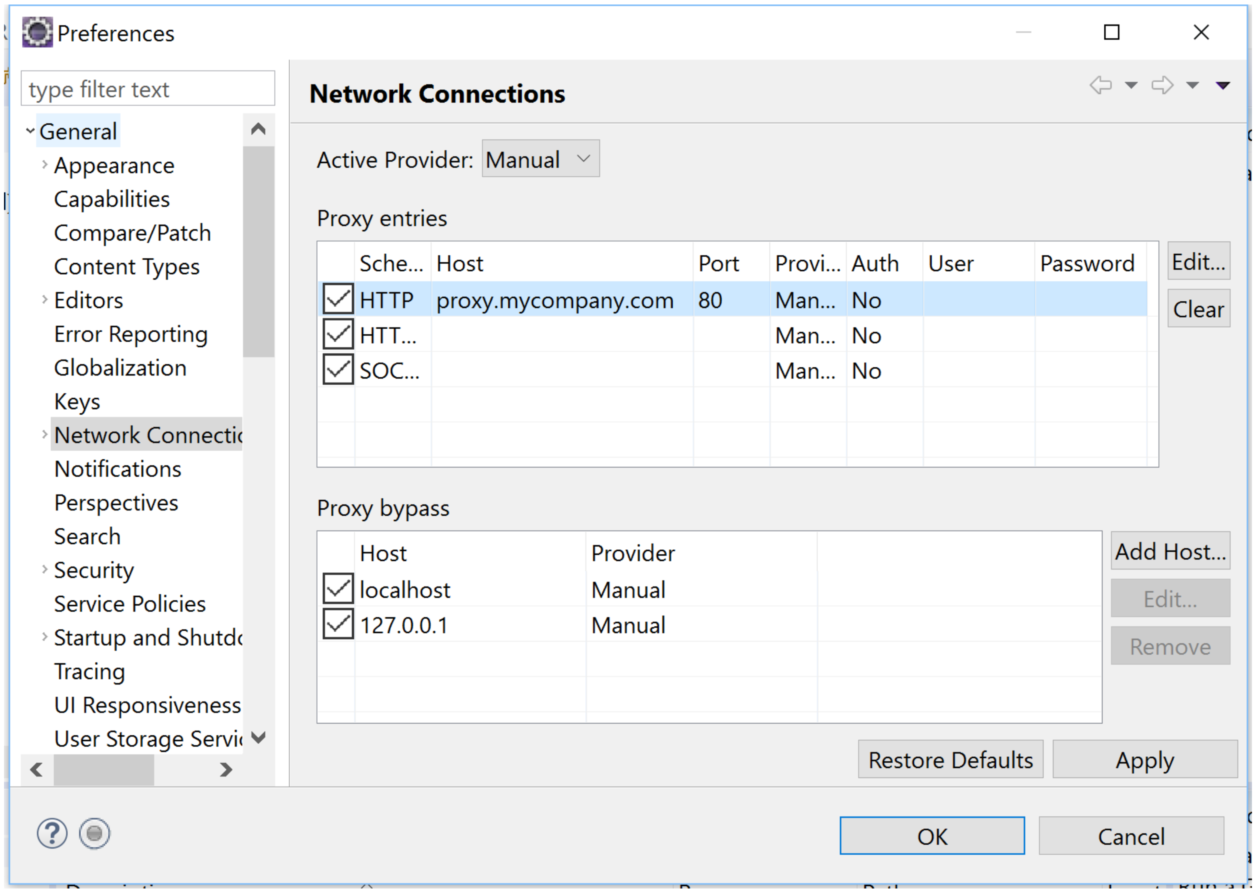
Name	Context	Description	Auto Ins
☑ @author	Javadoc	author name	on
☑ <b>	Javadoc	<b></b>	on
☑ <code>	Javadoc	<code></code>	on
☑ <i>	Javadoc	<i></i>	on
☑ <pre>	Javadoc	<pre></pre>	on
☑ active_task	Javadoc	active task	on
☑ addlistener	SWT statements	add a listener to a Widget	
☑ arrayadd	Java statements	add an element to an array	
☑ arraymerge	Java statements	merge two arrays into one	
☑ Browser	SWT statements	new Browser	
☑ Button	SWT statements	new Button	
☑ cast	Java statements	dynamic cast	
☑ catch	Java statements	catch block	
☑ Combo	SWT statements	new Combo	
☑ Composite	SWT statements	new Composite with GridL...	

☑ Use code formatter

Setting Proxy Requirements for Liferay IDE

If you have proxy server requirements and want to configure your http(s) proxy to work with Liferay IDE, follow the instructions below.

1. Navigate to Eclipse's *Window* → *Preferences* → *General* → *Network Connections* menu.
2. Set the *Active Provider* drop-down selector to *Manual*.
3. Under *Proxy entries*, configure both proxy HTTP and HTTPS by clicking the field and selecting the *Edit* button.


Preferences

Network Connections

Active Provider: Manual

Proxy entries

	Sche...	Host	Port	Provi...	Auth	User	Password
✓	HTTP	proxy.mycompany.com	80	Man...	No		
✓	HTT...			Man...	No		
✓	SOC...			Man...	No		

Proxy bypass

	Host	Provider
✓	localhost	Manual
✓	127.0.0.1	Manual

Figure 1: You can configure your proxy settings in IDE's Network Connections menu.

4. For each schema (HTTP and HTTPS), enter your proxy server's host, port, and authentication settings (if necessary).

Note: Do not leave whitespace at the end of your proxy host or port settings.

5. Once you've configured your proxy entry, click *OK* → *OK*.

If you're working with a Liferay Workspace in IDE, you'll need to configure your proxy settings for that environment too. See the Setting Proxy Requirements for Liferay Workspace for more details.

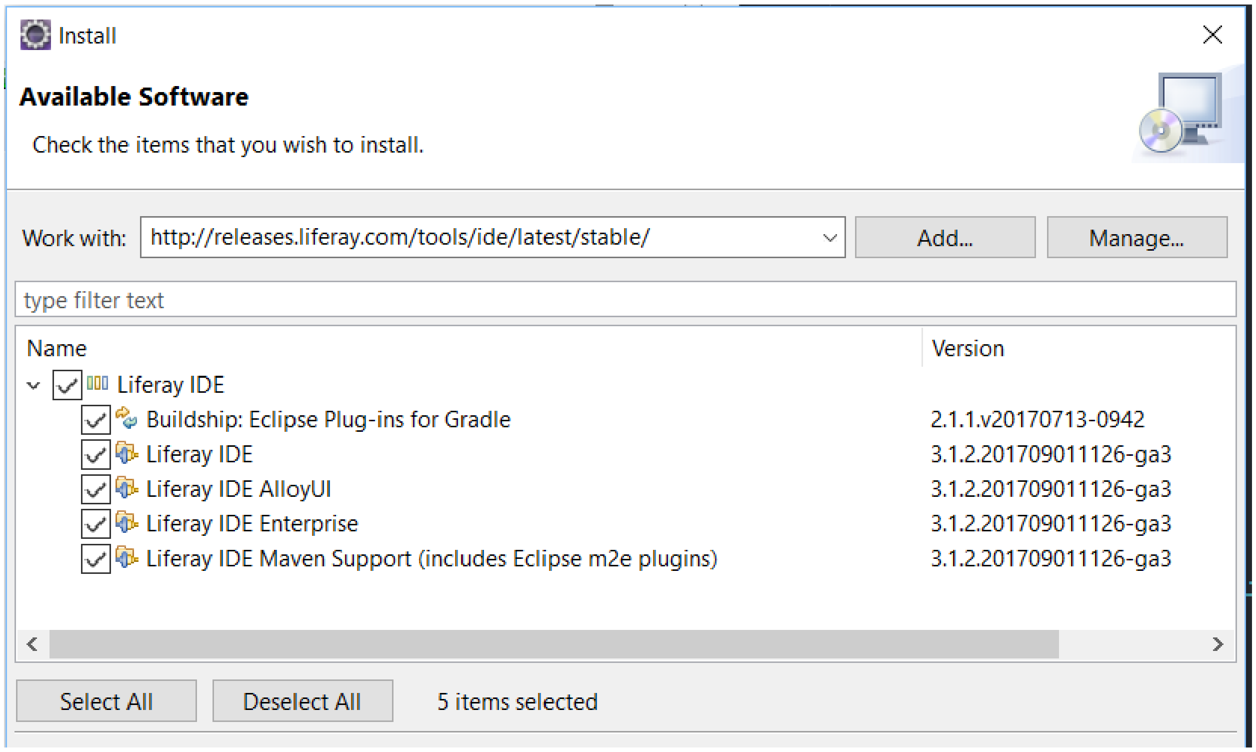
Awesome! You've successfully configured Liferay IDE's proxy settings!

Updating Liferay IDE

If you're already using Liferay IDE but need to update your environment, follow the steps below:

1. In Liferay IDE, go to *Help* → *Install New Software....*
2. In the *Work with* field, copy in the URL http://releases.liferay.com/tools/ide/latest/stable/.

3. You'll see the IDE components in the list below. Check them off and click *Next*.



Install

Available Software

Check the items that you wish to install.

Work with: http://releases.liferay.com/tools/ide/latest/stable/

Name	Version
✓ Liferay IDE	
✓ Buildship: Eclipse Plug-ins for Gradle	2.1.1.v20170713-0942
✓ Liferay IDE	3.1.2.201709011126-ga3
✓ Liferay IDE AlloyUI	3.1.2.201709011126-ga3
✓ Liferay IDE Enterprise	3.1.2.201709011126-ga3
✓ Liferay IDE Maven Support (includes Eclipse m2e plugins)	3.1.2.201709011126-ga3

5 items selected

Figure 1: Make sure to check all the IDE components you wish to install.

4. Accept the terms of the agreements. Click *Next*, and IDE is updated. You must restart IDE for the updates to take effect.

You're now on the latest version of Liferay IDE!

Deploying Modules with Liferay IDE

Deploying modules in Liferay IDE is a cinch. Before deploying your module, make sure you have a Liferay server configured in IDE. To see how to do this, see the Installing a Server in Liferay IDE

There are two ways to deploy a module to your Liferay server. You should start your Liferay server before attempting to deploy to it.

1. Select the module from the Package Explorer window and drag it to your Liferay server in the Servers window.

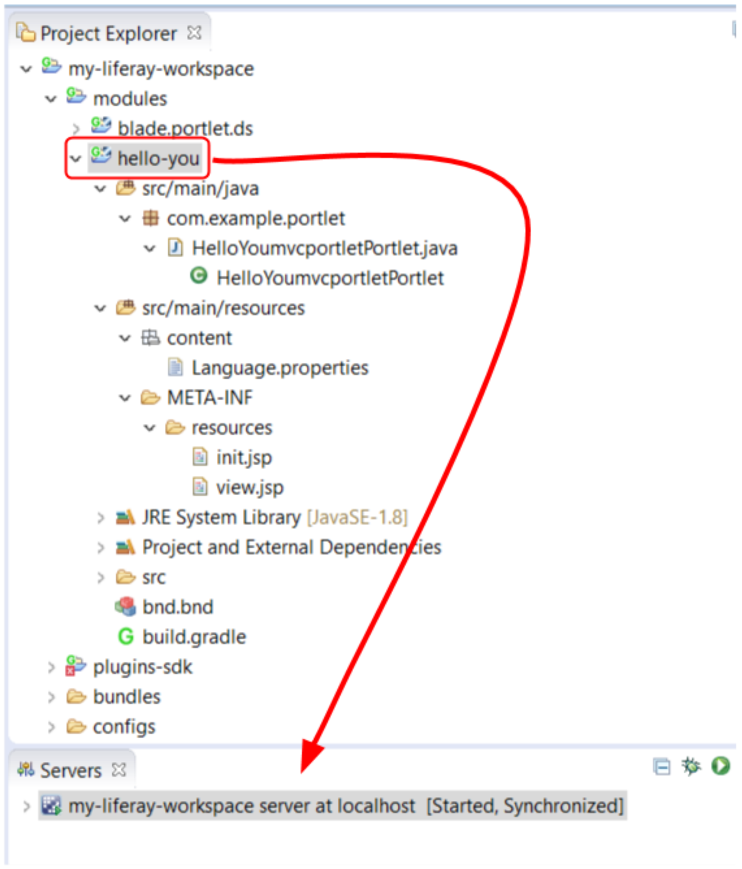
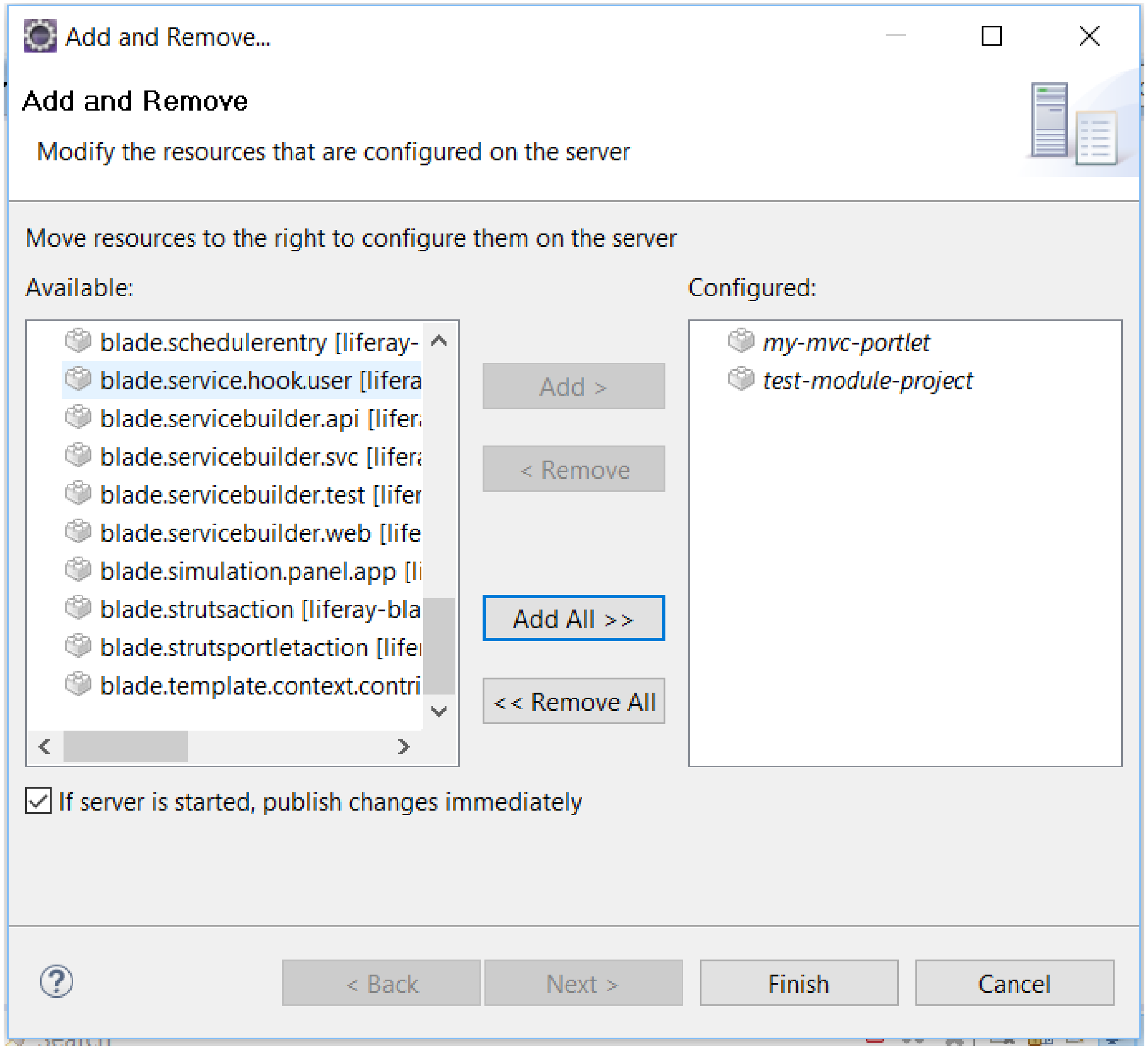


Figure 1: You can use the drag-and-drop method to deploy your module to Liferay Portal.

2. Right-click the server from the Servers window and select *Add and Remove…*. Add the module(s) you'd like to deploy from the Available window to the Configured window. Then click *Finish*.


Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

- blade.schedulerentry [liferay-
- blade.service.hook.user [lifera
- blade.servicebuilder.api [lifera
- blade.servicebuilder.svc [lifera
- blade.servicebuilder.test [lifer
- blade.servicebuilder.web [life
- blade.simulation.panel.app [li
- blade.strutsaction [liferay-bla
- blade.strutsportletaction [lifer
- blade.template.context.contri

Configured:

- *my-mvc-portlet*
- *test-module-project*

☑ If server is started, publish changes immediately

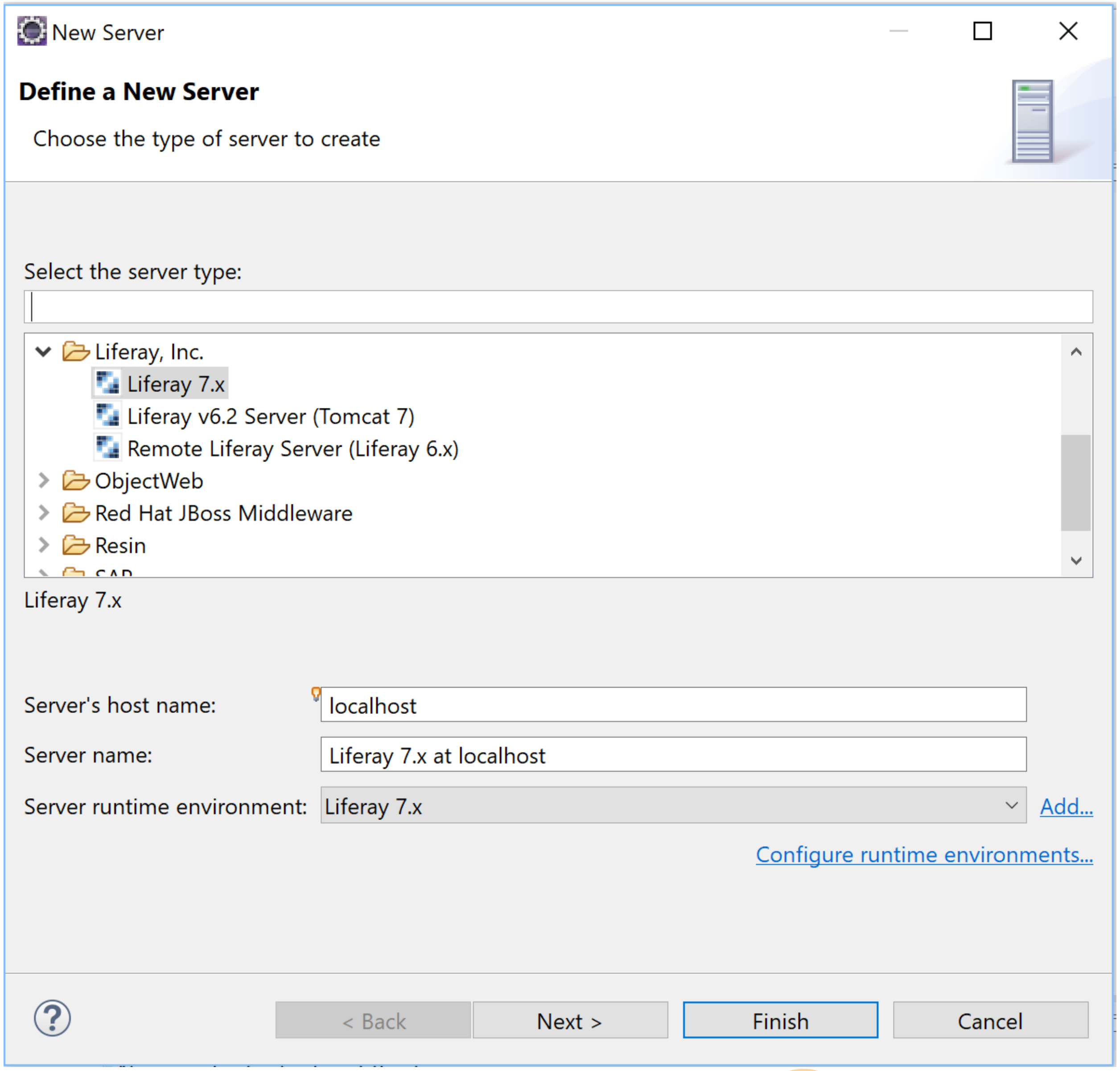
Figure 2: Using the this deployment method is convenient when deploying multiple module projects.

Installing a Server in Liferay IDE

Installing a server in Liferay IDE is easy. In just a few steps you'll have your server up and running. Follow these steps to install your server:

1. In the Servers view, click the *No Servers are available* link. If you already have a server installed, you can install a new server by right-clicking in the Servers view and selecting *New* → *Server*. This brings up a wizard that walks you through the process of defining a new server.

2. Select the type of server you would like to create from the list of available options. For a standard server, open the *Liferay, Inc.* folder and select the *Liferay 7.x* option. You can change the server name to something more unique too; this is the name displayed in the Servers view. Then click *Next*. If you're creating a server for the first time, continue to the next step.



New Server

Define a New Server

Choose the type of server to create

Select the server type:

Liferay, Inc.
- Liferay 7.x
- Liferay v6.2 Server (Tomcat 7)
- Remote Liferay Server (Liferay 6.x)

ObjectWeb

Red Hat JBoss Middleware

Resin

Liferay 7.x

Server's host name: localhost

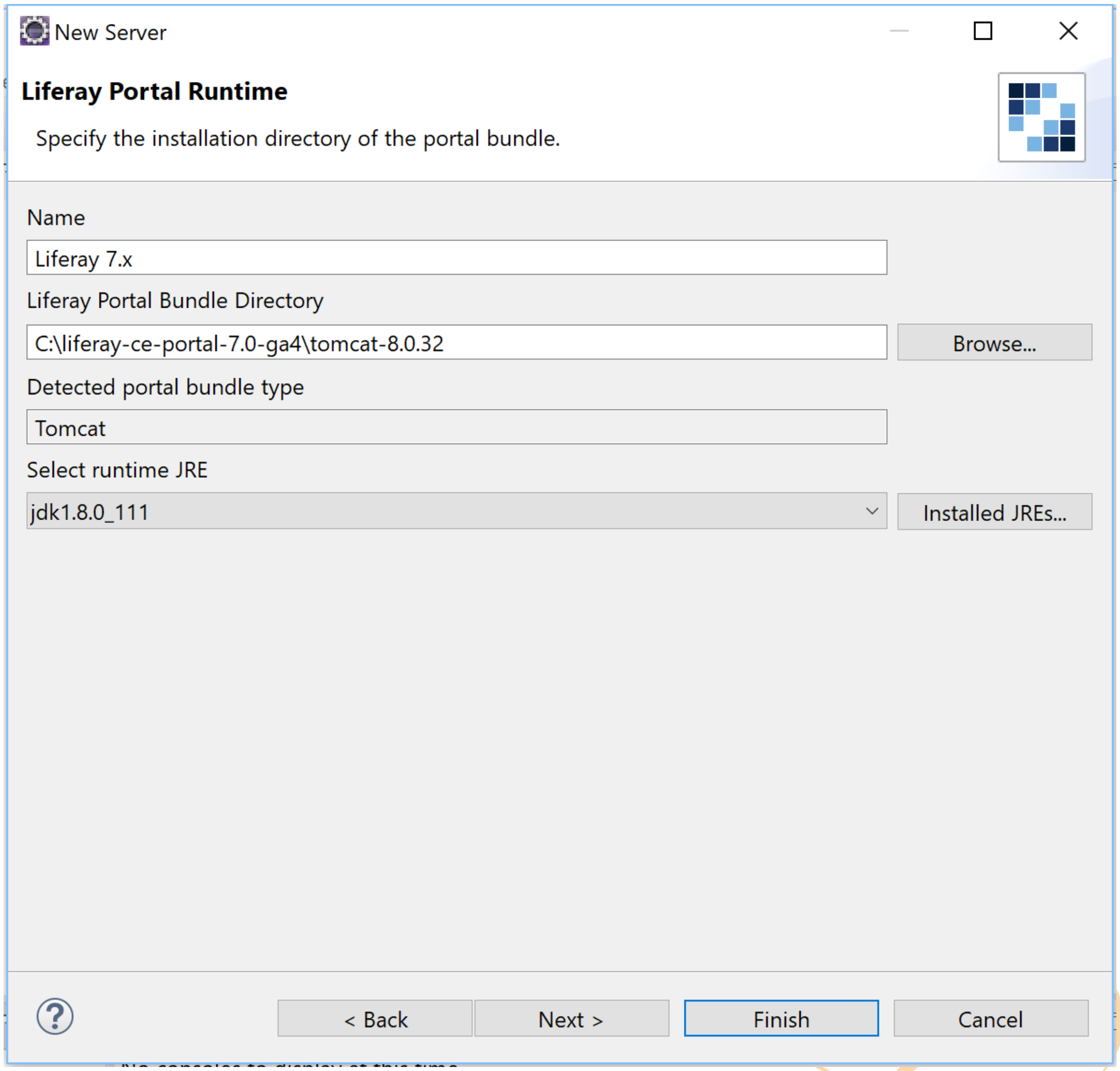
Server name: Liferay 7.x at localhost

Server runtime environment: Liferay 7.x

Figure 1: Choose the type of server you want to create.

Note: If you've already configured previous Liferay servers, you'll be provided the *Server runtime environment* field, which lets you choose previously configured runtime environments. If you want to re-add an existing server, select one from the dropdown menu. You can also add a new server by selecting *Add*, or you can edit existing servers by selecting *Configure runtime environments*. Once you've configured the server runtime environment, select *Finish*. If you selected an existing server, your server installation is finished; you can skip steps 3-5.

3. Enter a name for your server. This is the name for the Liferay Portal runtime configuration used by IDE. This is not the display name used in the Servers tab.
4. Browse to the installation folder of the Liferay Portal bundle. For example, C:\liferay-ce-portal-7.0-ga4\tomcat-8.0.32.


New Server

Liferay Portal Runtime

Specify the installation directory of the portal bundle.

Name: Liferay 7.x

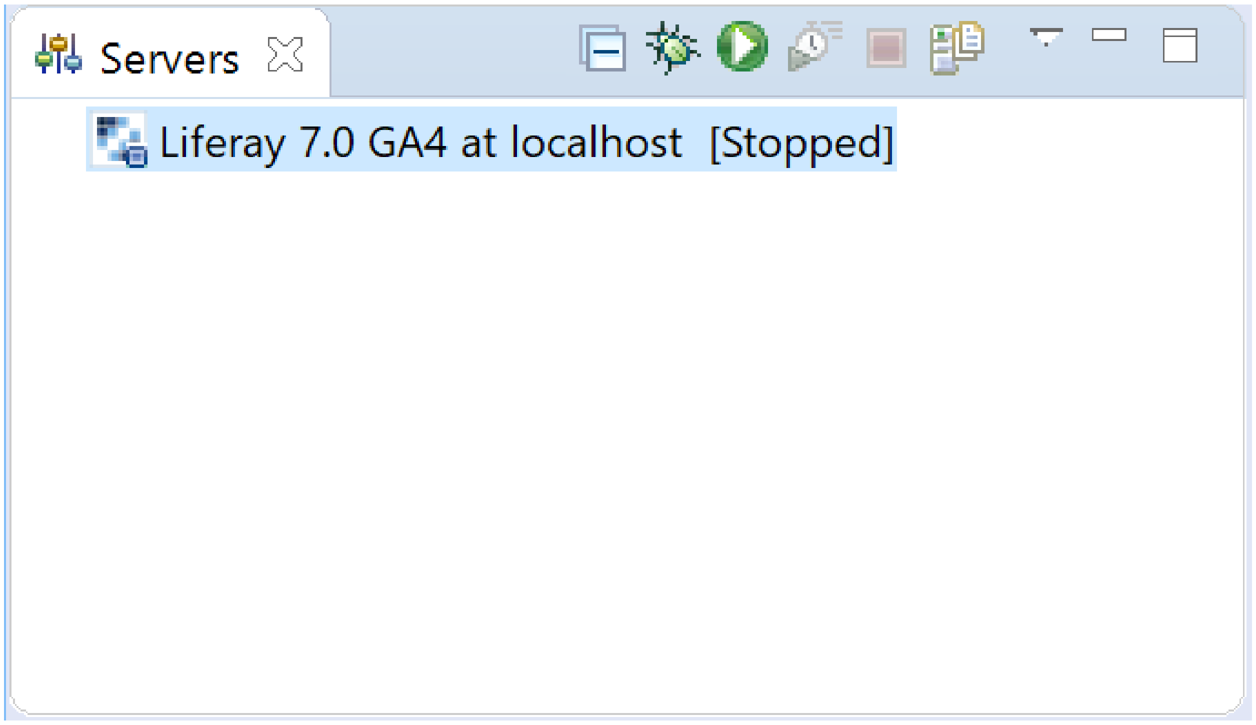
Liferay Portal Bundle Directory: C:\liferay-ce-portal-7.0-ga4\tomcat-8.0.32

Detected portal bundle type: Tomcat

Select runtime JRE: jdk1.8.0_111

Figure 2: Specify the installation folder of the bundle.

5. Select a runtime JRE and click *Finish*. Your new server appears under the Servers view.



Liferay 7.0 GA4 at localhost [Stopped]

Figure 3: Your new server appears under the *Servers* view.
Your server is now available in Liferay IDE!
For reference, here's how the IDE server buttons work with your Liferay Portal instance:

- *Start* ( ): Starts the server.
- *Stop* ( ): Stops the the server.
- *Debug* ( ): Starts the server in debug mode. For more information on debugging in Eclipse, see the Eclipse Debugging article.

Now you're ready to use your server in Liferay IDE!

SEARCH CLASS HIERARCHY

Inspecting classes that extend a similar superclass can help you find useful patterns and examples for how you can develop your own app. For example, suppose your app extends the MVCPortlet class. You an search classes that extend that same class in IDE by right-clicking the MVCPortlet declaration and selecting *Open Type Hierarchy*. This opens a window that lets you inspect all classes residing in the target platform that extend MVCPortlet.

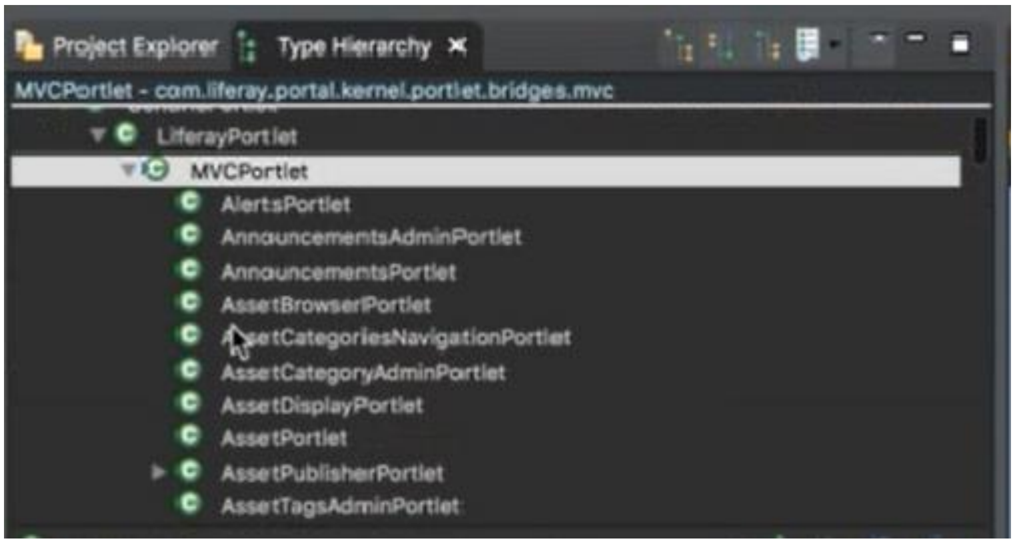


Figure 1: Browse the Type Hierarchy window and open the provided classes for examples on how to extend a class.

SEARCH METHOD DECLARATIONS

Sometimes you want a search to be more granular, exploring the declarations of a specific method provided by a class/interface. Liferay IDE's advanced search has no limits; Liferay Workspace's target platform indexing provides method exploration too!

Suppose in the MVCPortlet class you're extending, you'd like to search for declarations of its doView method you're overriding. You can do this by right-clicking the doView method declaration in your custom app's class and selecting *Declarations* → *Workspace*.

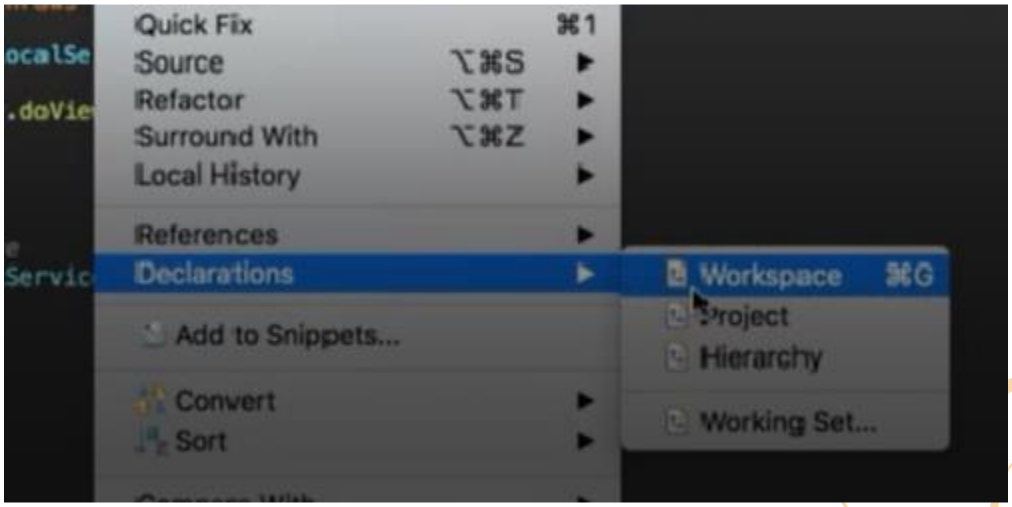
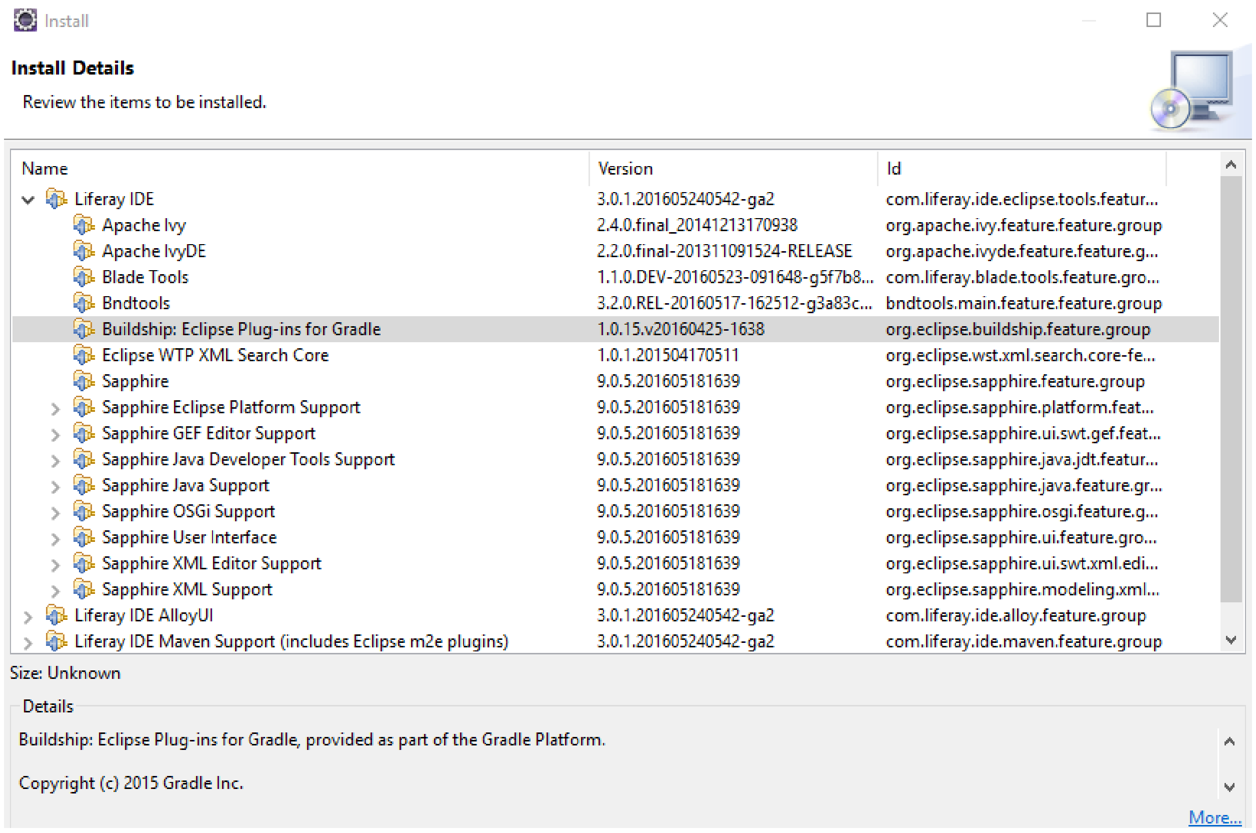


Figure 2: All declarations of the method are returned in the Search window

Using Gradle in Liferay IDE

Gradle is a popular open source build automation system. You can take full advantage of Gradle in Liferay IDE by utilizing Buildship, which is a collection of Eclipse plugin-ins that provide support for building software using Gradle with Liferay IDE. Buildship is bundled with Liferay IDE versions 3.0 and higher.



Name	Version	Id
Liferay IDE	3.0.1.201605240542-ga2	com.liferay.ide.eclipse.tools.featur...
Apache Ivy	2.4.0.final_20141213170938	org.apache.ivy.feature.feature.group
Apache IvyDE	2.2.0.final-201311091524-RELEASE	org.apache.ivyde.feature.feature.g...
Blade Tools	1.1.0.DEV-20160523-091648-g5f7b8...	com.liferay.blade.tools.feature.gro...
Bndtools	3.2.0.REL-20160517-162512-g3a83c...	bndtools.main.feature.feature.group
Buildship: Eclipse Plug-ins for Gradle	1.0.15.v20160425-1638	org.eclipse.buildship.feature.group
Eclipse WTP XML Search Core	1.0.1.201504170511	org.eclipse.wst.xml.search.core-fe...
Sapphire	9.0.5.201605181639	org.eclipse.sapphire.feature.group
Sapphire Eclipse Platform Support	9.0.5.201605181639	org.eclipse.sapphire.platform.feat...
Sapphire GEF Editor Support	9.0.5.201605181639	org.eclipse.sapphire.ui.swt.gef.feat...
Sapphire Java Developer Tools Support	9.0.5.201605181639	org.eclipse.sapphire.java.jdt.featur...
Sapphire Java Support	9.0.5.201605181639	org.eclipse.sapphire.java.feature.gr...
Sapphire OSGi Support	9.0.5.201605181639	org.eclipse.sapphire.osgi.feature.g...
Sapphire User Interface	9.0.5.201605181639	org.eclipse.sapphire.ui.feature.gro...
Sapphire XML Editor Support	9.0.5.201605181639	org.eclipse.sapphire.ui.swt.xml.edi...
Sapphire XML Support	9.0.5.201605181639	org.eclipse.sapphire.modeling.xml...
Liferay IDE AlloyUI	3.0.1.201605240542-ga2	com.liferay.ide.alloy.feature.group
Liferay IDE Maven Support (includes Eclipse m2e plugins)	3.0.1.201605240542-ga2	com.liferay.ide.maven.feature.group

Size: Unknown

Details

Buildship: Eclipse Plug-ins for Gradle, provided as part of the Gradle Platform.

Copyright (c) 2015 Gradle Inc.

Figure 1: Navigate to *Help* → *Installation Details* to view plugins included in Liferay IDE.

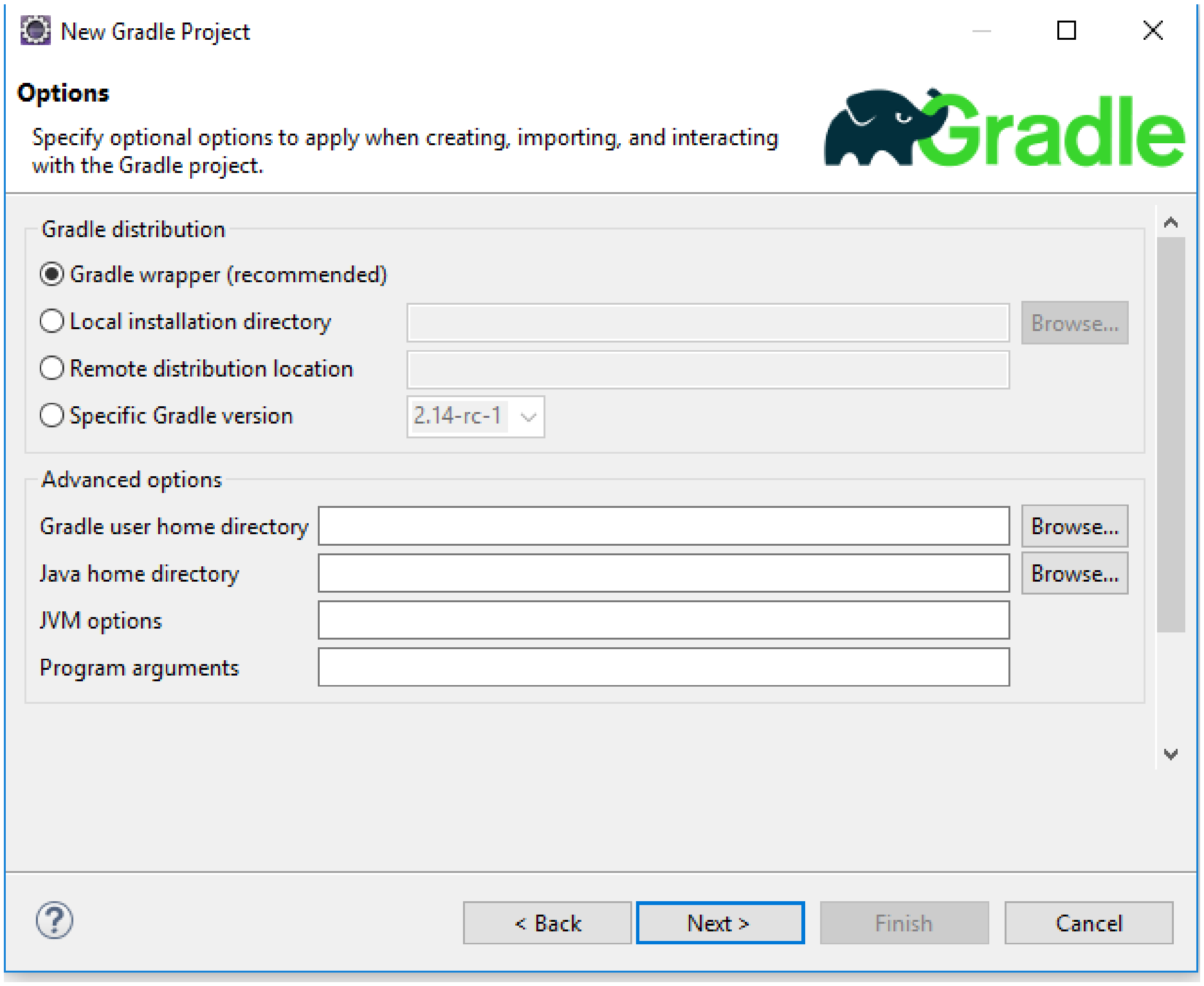
The first thing you'll learn about in this tutorial is creating Gradle projects in IDE.

CREATING AND IMPORTING GRADLE PROJECTS

You can create a Gradle project by using the Gradle Project wizard.

1. Navigate to *File* → *New* → *Project...* and select *Gradle* → *Gradle Project*. Finally, click *Next* → *Next*.
2. Enter a valid project name. You can also specify your project location and working sets.

3. Optionally, you can navigate to the next page and specify your Gradle distribution and other advanced options. Once you're finished, select *Finish*.


New Gradle Project

Options

Specify optional options to apply when creating, importing, and interacting with the Gradle project.

Gradle distribution

(●) Gradle wrapper (recommended)
( ) Local installation directory
( ) Remote distribution location
( ) Specific Gradle version: 2.14-rc-1

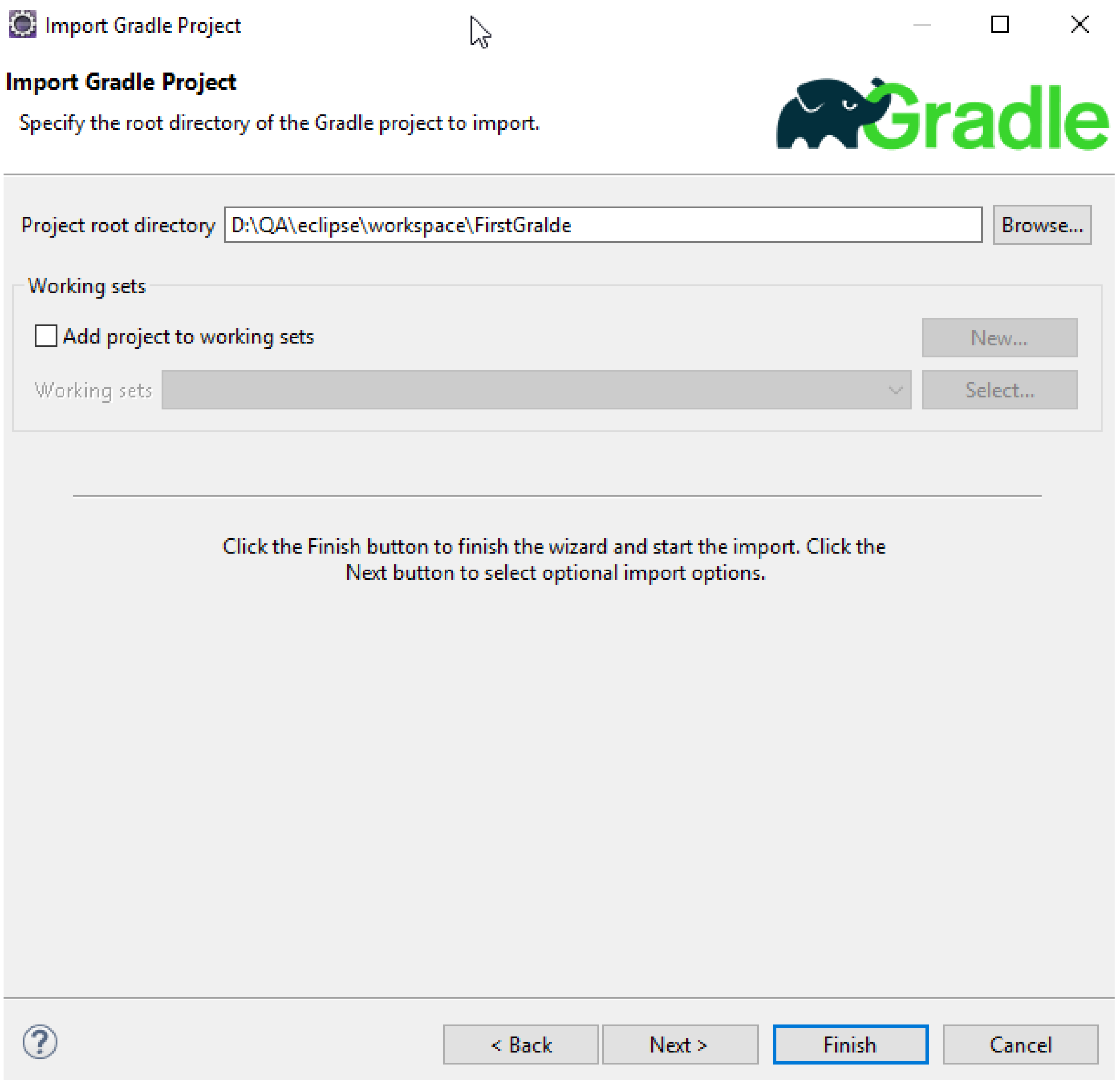
Advanced options

Gradle user home directory
Java home directory
JVM options
Program arguments

Figure 2: You can specify your Gradle distribution and advanced options such as home directories, JVM options, and program arguments.

You can also import existing Gradle projects in Liferay IDE.

1. Go to *File → Import... → Gradle → Gradle Project → Next → Next.*


Import Gradle Project

Import Gradle Project

Specify the root directory of the Gradle project to import.

Project root directory: D:\QA\eclipse\workspace\FirstGralde

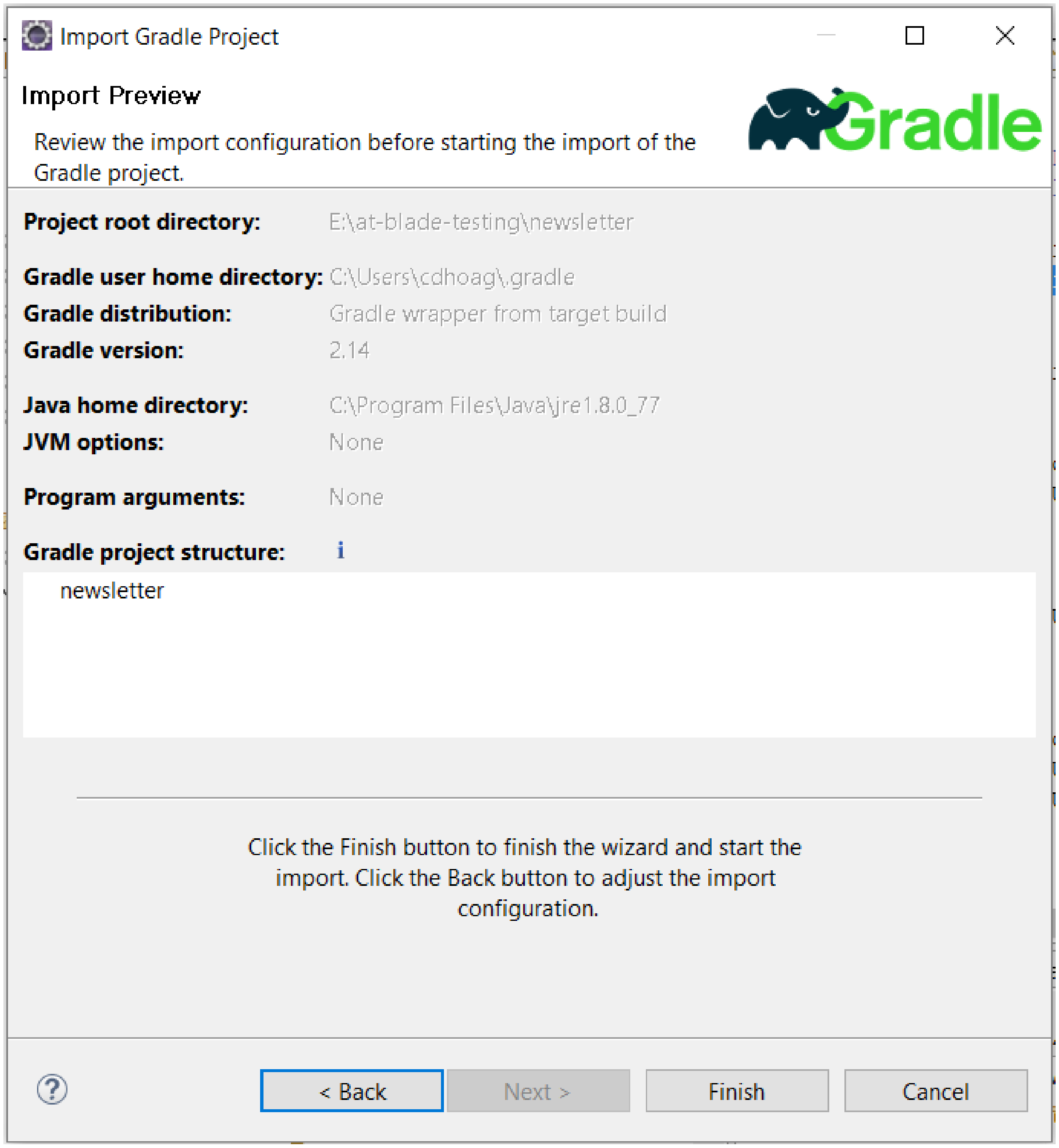
Working sets

Add project to working sets

Click the Finish button to finish the wizard and start the import. Click the Next button to select optional import options.

Figure 3: You can specify what Gradle project to import from the *Import Gradle Project* wizard.

2. Click the *Browse…* button to choose a Gradle project.
3. Optionally, you can navigate to the next page and specify your Gradle distribution and other advanced options. Once you're finished, click *Next* again to review the import configuration. Select *Finish* once you've confirmed your Gradle project import.


Import Gradle Project

Import Preview

Review the import configuration before starting the import of the Gradle project.

Project root directory:	E:\at-blade-testing\newsletter
Gradle user home directory:	C:\Users\cdhoag\.gradle
Gradle distribution:	Gradle wrapper from target build
Gradle version:	2.14
Java home directory:	C:\Program Files\Java\jre1.8.0_77
JVM options:	None
Program arguments:	None
Gradle project structure:	

newsletter

Click the Finish button to finish the wizard and start the import. Click the Back button to adjust the import configuration.

Figure 4: You can preview your Gradle project's import information.

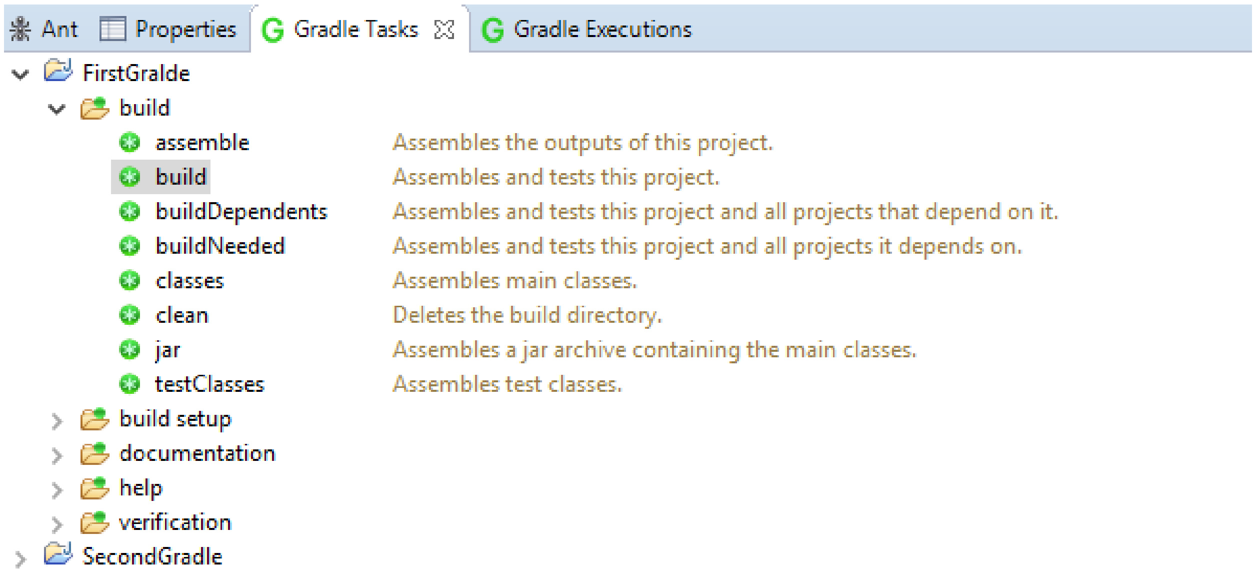
Next you'll learn about Gradle tasks and executions, and learn how to run and display them in Liferay IDE.

GRADLE TASKS AND EXECUTIONS

Liferay IDE provides two views to enhance your developing experience using Gradle: Gradle Tasks and Gradle Executions. You can open these views by following the instructions below.

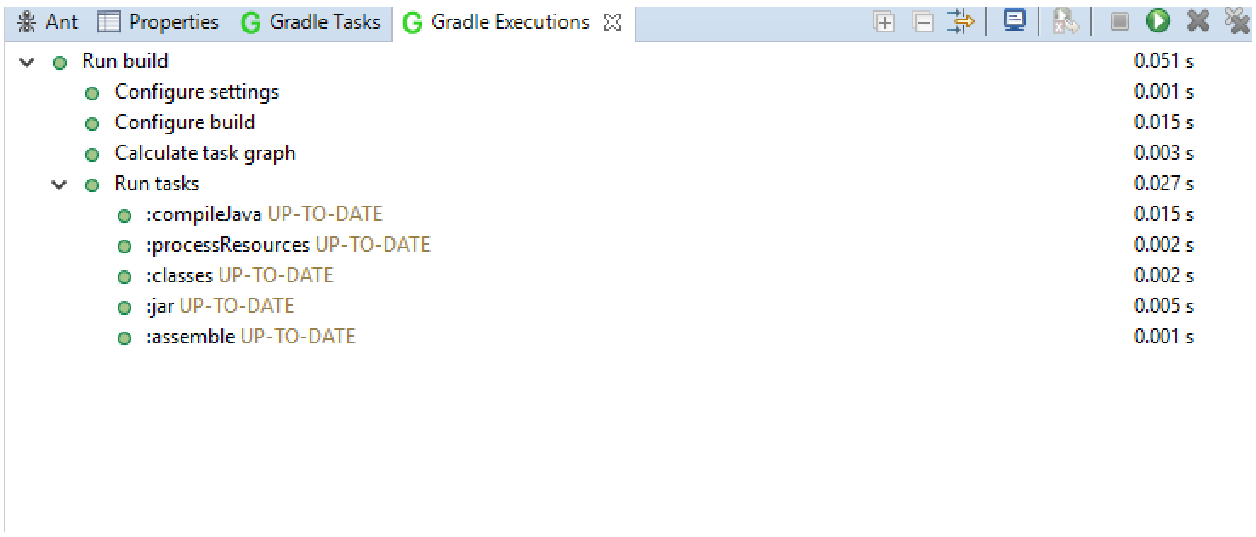
1. Go to *Window* → *Show View* → *Other*….
2. Navigate to the *Gradle* folder and open *Gradle Tasks* and *Gradle Executions*.

Gradle tasks and executions views open automatically once you create or import a Gradle project.
The Gradle Tasks view allows you to display the Gradle tasks available for you to use in your Gradle project. Users can execute a task listed under the Gradle project by double-clicking it.



```
FirstGralde
  build
    assemble          Assembles the outputs of this project.
    build             Assembles and tests this project.
    buildDependents   Assembles and tests this project and all projects that depend on it.
    buildNeeded       Assembles and tests this project and all projects it depends on.
    classes           Assembles main classes.
    clean             Deletes the build directory.
    jar               Assembles a jar archive containing the main classes.
    testClasses       Assembles test classes.
  build setup
  documentation
  help
  verification
SecondGradle
```

Figure 5: Navigate into your preferred Gradle project to view its available Gradle tasks.
Once you’ve executed a Gradle task, you can open the Gradle Executions view to inspect its output.



```
Run build                                  0.051 s
  Configure settings                       0.001 s
  Configure build                          0.015 s
  Calculate task graph                     0.003 s
  Run tasks                                0.027 s
    :compileJava UP-TO-DATE                0.015 s
    :processResources UP-TO-DATE           0.002 s
    :classes UP-TO-DATE                    0.002 s
    :jar UP-TO-DATE                        0.005 s
    :assemble UP-TO-DATE                   0.001 s
```

Figure 6: The Gradle Executions view helps you visualize the Gradle build process.

Keep in mind that if you change the Gradle build scripts inside your Gradle projects (e.g., build.gradle or settings.gradle), you must refresh the project so Liferay IDE can account for the change and display it properly in your views. To refresh a Gradle project, right-click on the project and select *Gradle* → *Refresh Gradle Project*.

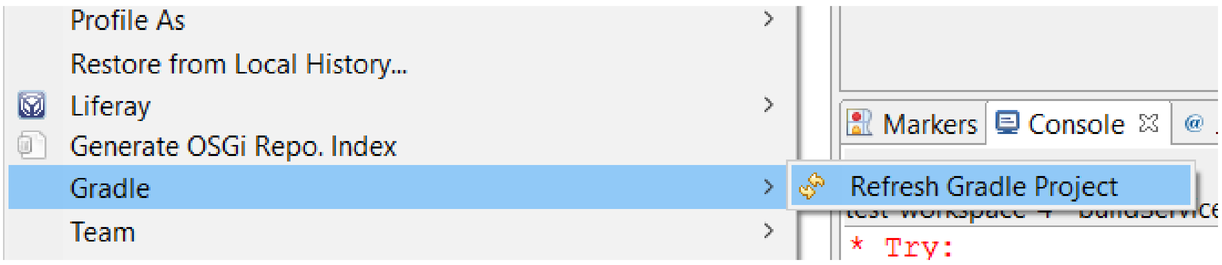


Figure 7: Make sure to always refresh your Gradle project in Liferay IDE after build script edits.

If you prefer Eclipse refresh your Gradle projects automatically, navigate to *Window* → *Preferences* → *Gradle* and enable the *Automatic Project Synchronization* checkbox. If you'd like to enable Gradle's automatic synchronization for just one Gradle project, you can right-click a Gradle project and select *Properties* → *Gradle* and enable auto sync that way. This feature is available in Buildship version 2.2+, so make sure you have the required version.

Build gradle project

```
./gradlew cleanEclipse eclipse
```

Git

R Click/ Replace/ Head revision
CTRL ALT H: call hierarchi
ALT SHIFT T: Refactor

Install doma

https://doma.readthedocs.io/en/stable/getting-started-eclipse/