

7 Projects

A project in jGRASP is essentially one or more files which may be located in the same or different directories. When a “project” is created, all information about the project, including project settings and file locations, is stored in a *project file* with the .gpj extension.

Although projects are not required to do simple operations such as Compile and Run, to generate UML class diagrams and to use many of the Object Workbench features, you must organize your Java files in a Project. UML Class Diagrams and the Object Workbench are discussed in Sections 5 and 6. Many users will find projects useful independent of the UML and Object Workbench features.

Before doing this tutorial, be sure you have read the tutorial entitled *Getting Started with Objects* since the concept of a jGRASP project is first introduced there.

Objectives – When you have completed this tutorial, you should be able to create projects, add files to them, remove files from them, generate documentation, and close projects.

The details of these objectives are captured in the hyperlinked topics listed below.

7.1 Creating a Project

7.2 Adding files to the Project

7.3 Removing files from the Project

7.4 Generating Documentation for the Project (Java only)

7.5 Jar File Creation and Extraction

7.6 Closing a Project

7.7 Exercises

7.1 Creating a Project

On the Desktop menu, click **Project > New > New Standard Project** (Figure 7-1) to open the **New Project** dialog. *Note that the “New J2ME Project” option should only be selected if you have installed the Java Wireless Tool Kit (WTK) and you plan to develop a project based on the Java 2 Micro Edition (J2ME).*

Within the New Project dialog (Figure 7-2), notice the two check boxes (*Add Files Now* and *Open UML Window*). Normally, you would want to have the *Add Files Now* checked ON so that as soon as you click the Create button, the Add Files dialog will pop up. If you are working in Java, you may also want to turn ON the *Open UML Window* option. This will generate the UML class diagram and open the UML window (see Section 5 for details).

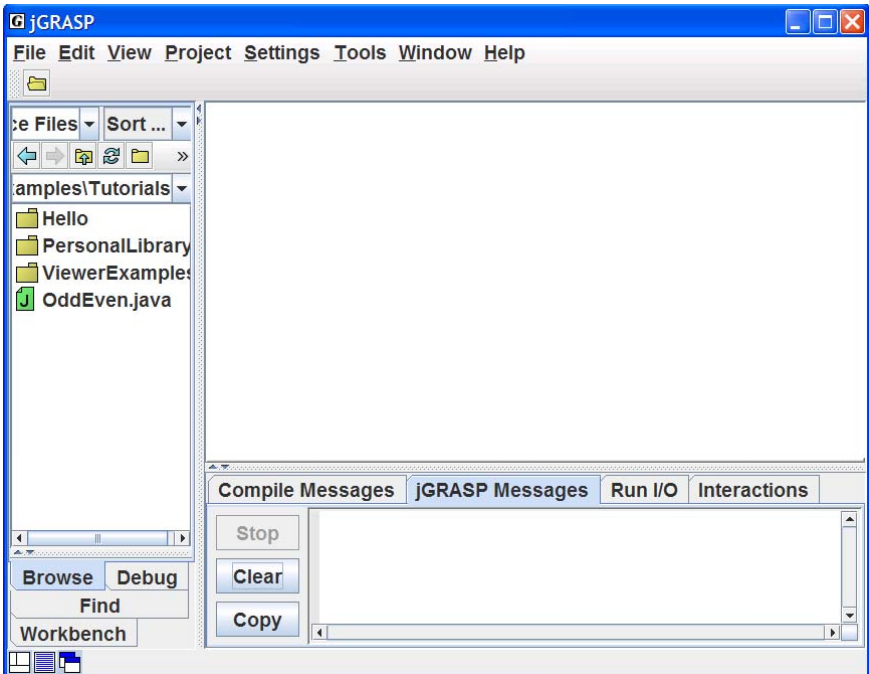


Figure 7-1. Creating a Project

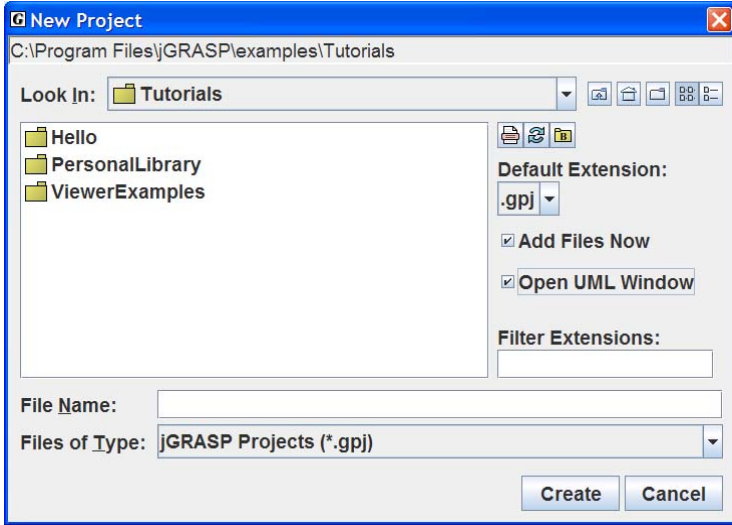


Figure 7-2. New Project dialog

Navigate to the directory where you want the project to reside and enter the project file name. It is recommended that the project file be stored in the same directory as the file containing *main*. A useful naming convention for a project is *ClassnameProject* where *Classname* is the name of the class that contains *main*. For example, since the *PersonalLibrary* class contains *main*, an appropriate name for the project file would be *PersonalLibraryProject*.

After entering the project file name, click **Create** to save the project file. Notice the new project file with *.gpj* extension is listed in the Files section of the Browse tab. The project is also listed in the Open Projects section of Browse tab. If *Add Files Now* was checked ON when you created the project, the Add Files dialog will pop up. As files are added to the project, they will appear under the project name in the Open Projects section of the Browse tab. When you have finished adding files, click the *Close button* on the dialog. You can always add more files to a project later.

Note that when you have multiple projects open, these are all listed in the Open Projects section of the Browse tab. If you open a UML window for one or more projects and/or if you open one or more CSD windows for files in projects, then the UML or CSD window with focus will determine which open project has focus. The project with focus will have a black square in the project symbol and the project name will be displayed in the title bar of the jGRASP desktop.

7.2 Adding files to the Project

The Browse tab is split to show the current file directory in the top part and the open projects in the lower part as shown in Figure 7-3. After a project has been created and/or opened, there are several ways to add Java files to the project.

- (1) **From Browse Tab** - Drag the file (left click and hold) from the Files section to the project in the Open Projects section below.
- (2) **From Browse Tab** - Drag the file from the Files section to the UML Window.
- (3) **In Browse Tab** - Right click on the file and select **Add to Project**. (Figure 7-3).
- (4) **From CSD window** > Click **Project** > **Add files**.

You can also select multiple files (holding down the control or shift key), and add or drag the highlighted files all at once. The files in the project are shown beneath the project name in the Open Projects section of the Browse tab. Double-clicking on the project name (or single-clicking on the “handle” in front of the project name) will open or close the list of files in the project.

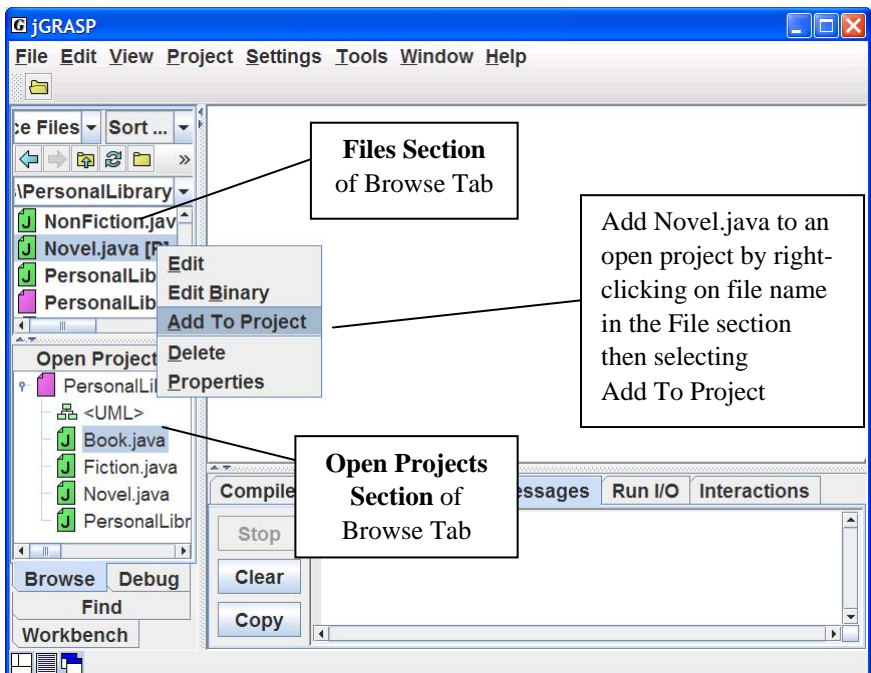


Figure 7-3. Adding a file to the Project

7.3 Removing files from the Project

You can remove files from the project by selecting one or more files in the Open Projects section of the Browse tab, then right clicking and selecting **Remove from Project(s)** as shown in Figure 7-4. You can also remove the selected file(s) by pressing *Delete* on the keyboard. Note that *removing* a file from a project does not delete the file from its directory, only from the project. However, you can delete a file by selecting it in the Files section of the Browse tab, then right-clicking and selecting **Delete** from the pop-up menu or by pressing the *Delete* key.

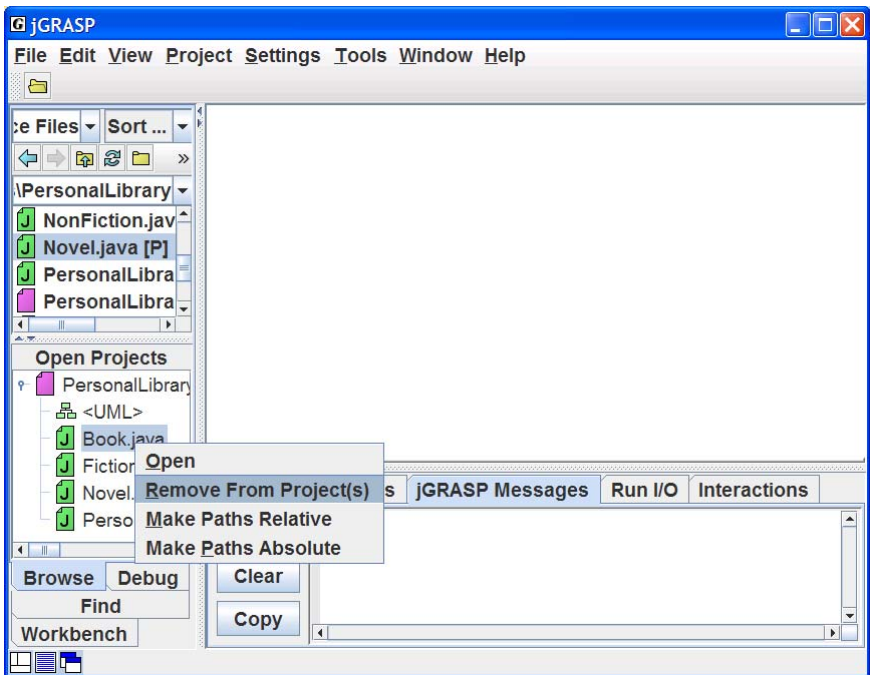


Figure 7-4. Removing a file from the Project

7.4 Generating Documentation for the Project (Java only)

Now that you have established a project, you have the option to generate project level documentation for your Java source code, i.e., application programmer interface (API) documentation. To generate the documentation for our example project, PersonalLibraryProject, select **Project** > **Generate Documentation** > **<PersonalLibraryProject>** as shown in the Figure 7-5. This will bring up the “Generate Documentation for Project” dialog which asks for the directory in which the generated HTML files are to be stored. The default directory name is the project name with “_doc” appended to it (e.g., PersonalLibraryProject_doc). Using the default name is recommended so that your documentation directories will have a standard naming convention. If the default directory is not indicated, click the **Default** button in the dialog. However, you are free to use any directory as the target. Click the **Generate** button on the dialog to start the process. jGRASP calls the javadoc utility, which is included with the JDK, to create a complete hyper-linked document.

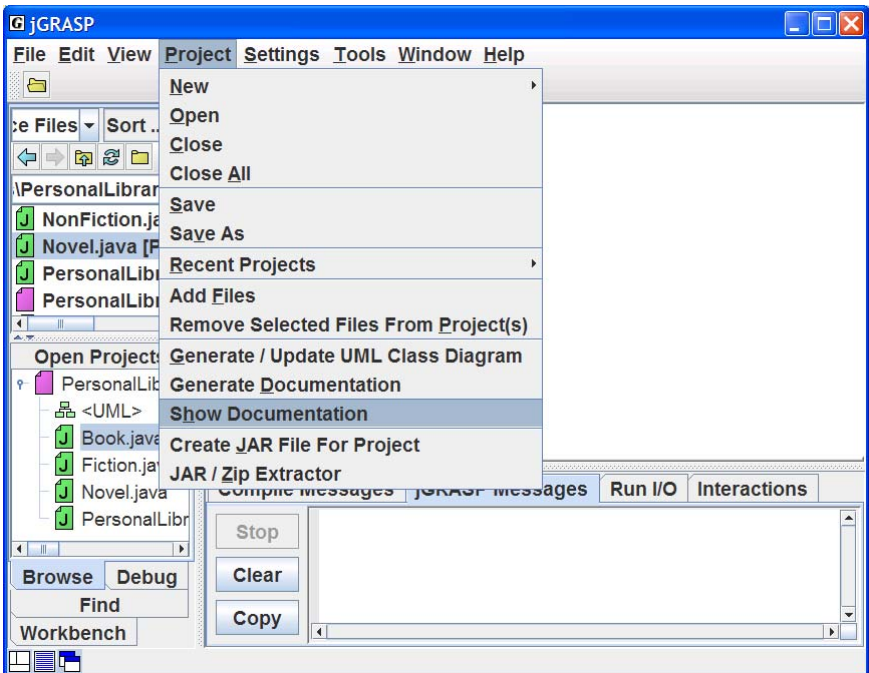


Figure 7-5. Generating Documentation for the Project

The documentation generated for PersonalLibraryProject is shown below in Figure 7-6. Note that in this example, even though no JavaDoc comments were included in the source file, the generated documentation is still quite useful. However, for even better documentation, JavaDoc formal comments should be included in the source code. When generated for a project, the documentation files are stored in a directory that becomes part of the project and, therefore, persists from one jGRASP session to the next. **Project > Show Documentation** can be used to display the documentation without regenerating it. However, if any changes have been made to a project source file and the file has been saved, jGRASP will indicate that the documentation needs to be regenerated. You may choose to view the documentation anyway or to regenerate the documentation.

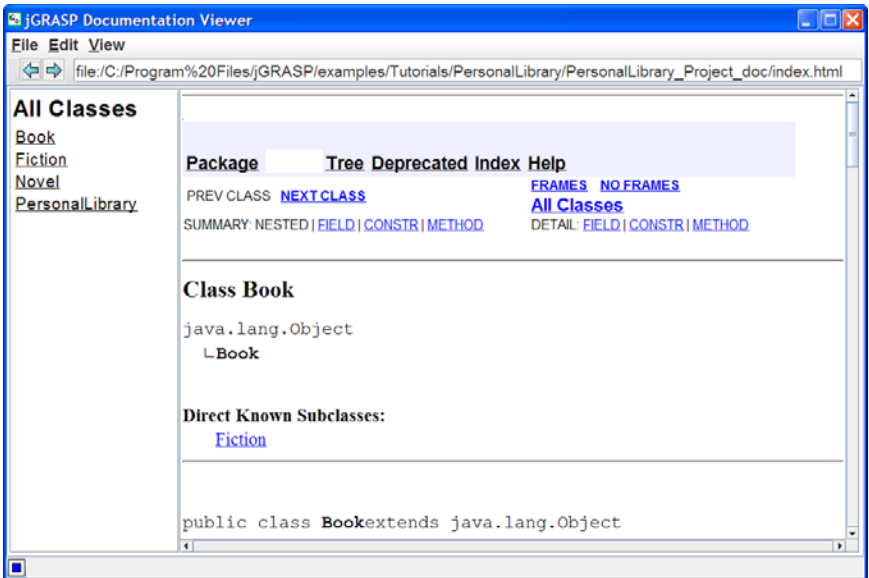


Figure 7-6. Project documentation

Documentation generated for an individual file is stored in a temporary directory for the duration of the jGRASP session unless the individual file is part of a project for which documentation has already been generated. In this case, Generate Documentation displays the existing documentation rather than generating a temporary documentation file.

7.5 Jar File Creation and Extraction

jGRASP provides a utility for the creation of and extraction of files from a Java Archive file (JAR) for your project. To create a JAR file, click **Project > Create Jar File for Project**. This will allow you to create a single compressed file containing your entire project.

The **Project > Jar/Zip Extractor** option enables you to extract the contents of a JAR or ZIP archive file.

These topics are described in more detail in jGRASP **Help** (find using the Index tab).

7.6 Closing a Project

When you exit jGRASP, the projects and files that are currently open on the desktop are remembered so that the next time you start jGRASP, you can pick up where you left off. However, to prevent clutter you should close the ones you are no longer using.

- (1) From the Desktop toolbar - Click **Project > Close All Projects**.
- (2) From the Desktop toolbar - Click **Project > Close** (if more than one project is open, select the one you want to close, e.g., *<PersonalLibraryProject>*).
- (3) From the Browse Tab – Right-click on the project file name in the Open Projects section of the Browse tab and select **Close**.

All project information is saved when you close the project as well as when you exit jGRASP. Note that closing a project does not close the files that are currently open. You can close these individually or all at once with **File – Close All Files**.

7.7 Exercises

- (1) Create a new project called `PersonalLibraryProject2` in the same directory folder as the original `PersonalLibraryProject`. During the create step, add the file `Book.java` to the new project. Close the Add Files dialog.
- (2) Add the other Java files in the directory to the project by dragging each file from the Files section of the Browse tab and dropping the files in `PersonalLibraryProject2` in the open projects section.
- (3) Remove a file from `PersonalLibraryProject2`. After verifying that the file was removed, add it back to the project.
- (4) Generate the documentation for `PersonalLibraryProject2`. After the Documentation Viewer pops up:
 - a. Click the Fiction class link in the API documentation (left side).
 - b. Click the Methods link to view the methods for the Fiction class.
 - c. Visit the other classes in the documentation for the project.
- (5) Close the project.
- (6) Open the project by double-clicking on the project file in the files section of the Browse tab.

Notes