

Setting Kelp properties

Table of Contents

1. JBuilder	1
PATH and CLASSPATH settings	1
Paths page	1
Output path	1
Source subtab	1
Required libraries subtab	2
Build properties	2
Generate source to output path option	2
Run options	3
Main class option	3
Application parameters option	3
2. Setting project properties in NetBeans	4
Classpath	4
Project directory	4
Source directory	4
Output directory	5
Deploy root	5
Run options	5
Setting up a project for use with the Kelp Application Wizard	5
3. Setting XMLC properties	6
XMLC project properties	6
XMLC node property page	7
How Kelp sets class names for XMLC	8
Setting XMLC options for selected files	9
Input Template node property page	9

Chapter 1. JBuilder

JBuilder provides many properties that let you customize your projects. After opening a project that you want to use with Enhydra, choose Project|Properties to configure path, compiler, and run settings. You do not need to enter anything in the Servlets tab to work with applications for Enhydra. Most of the settings you will use are found on the Paths tab.

Property pages are dialog boxes in which you set build options for the entire project or for a selected node in JBuilder. A node is an object in the project tree of a JBuilder project. For the purposes of this section, a node refers to a file in a project. Project property pages appear as tabs in the Project Properties dialog box. Node property pages let you customize options for a specific file in your project.

The Kelp project property pages add to JBuilder's existing Code Style, Paths, Run, and Compiler property pages. The Paths property page is where you set the library files your project requires.

- To open the Project Properties dialog box, choose Project|Project Properties.
- To open a Node property page, right-click a file in your project and choose Properties from the right-click menu.

JBuilder opens a property page specific to the file you have selected. If you right-clicked a Java source file, the property page will have RMI (Remote Method Invocation) and JNI (Java Native Interface) settings. To open the Input Template node property page, right-click any file with a .in extension in the template directory, /input. When you right-click an HTML file and select Properties, the XMLC node property page appears.

PATH and CLASSPATH settings

The PATH and CLASSPATH settings are critical for being able to compile and run your applications

Paths page

Configuring your project to use the correct paths for file output, project files, and libraries will simplify the development of applications for Enhydra. This section describes the options on the Project Properties Paths page that are relevant to using Kelp.

Output path

The output directory specifies where you want class files to be created. It is also where the Enhydra server looks for images that have relative references to the presentation objects. For example, the Kelp Application Wizard creates images under:

```
<source_directory>/<package_directory>/presentation/media
```

When you build the project inside JBuilder, the image is copied to the output directory along with the compiled class files. For the image to be displayed properly, the output directory must be set to a classes subdirectory under the source directory.

The Kelp Deployer also uses this setting when it creates deployable archives.

Source subtab

The source path is where the compiler looks for Java files and packages to compile. When you compile package names, the source files will be in subdirectories under the source directory. For example, if your highest-level package is com, the com directory will be located directly under a source path.

You can set the Source path to the same location as the project file. When you create a project using the Project wizard in JBuilder Foundation, it creates a directory for your project under jbprojects in your home directory. You can

use this directory as your source.

JBuilder lets you select multiple source paths. This can be useful when you are working on several modules from source code. For example, if you have checked out the Enhydra source code from CVS, you'll see that it is divided into several modules. Each module contains its own source directory.

Note: If you are using automatic source packages in JBuilder 5, make sure that your output path is not a subdirectory of any of your source paths.

Note: Remove any source paths that your project does not require.

Required libraries subtab

The application CLASSPATH must contain the following JARs in order to build and run Enhydra applications:

- `<Enhydra5.1_root>/lib/enhydra.jar`
- `<Enhydra5.1_root>/lib/xmlc.jar`
- `<Enhydra5.1_root>/lib/xerces.jar`
- `<Enhydra5.1_root>/lib/tomcat.jar`
- `<Enhydra5.1_root>/lib/core.jar`
- `<Enhydra5.1_root>/lib/xhtml.jar`
- `<Enhydra5.1_root>/lib/wireless.jar`
- `<Enhydra5.1_root>/lib/build/toolbox.jar`

You can add JARs to an application CLASSPATH by using JBuilder libraries. If you installed Kelp using the Windows installer, you already have an Enhydra library defined for you. If you are using JDBC in your application, you will need to add the JDBC driver as a library.

If you need to define an Enhydra library in JBuilder Foundation, click Add and then click New. Name your library and click Add to navigate to the JAR file you want to add.

Note: You may need to modify the Enhydra library if you have upgraded to a newer version of Enhydra. Make sure the JARs specified for the library match the JARs referenced in the IDE CLASSPATH. Use the About button in the XML Compiler dialog to determine the version of Enhydra that is in the IDE CLASSPATH.

Build properties

This section describes the options on the Project Properties Build page that are relevant to using Kelp.

Generate source to output path option

On the Build page, you can select Generate Source To Output Path to keep the generated XMLC Java files separate from your HttpPresentation implementations. If you select this option, the Java source files for the DOM classes will be created in a Generated Source directory under your output classes directory.

Run options

This section describes the options on the Project Properties Run page that are relevant to using Kelp.

Main class option

JBuilder Foundation lets you specify which class to run when you run the project. This class does not need to be part of your project's source files.

If you have an Enhydra library selected under the Required Libraries option, you can use the main class to launch the Enhydra server. Use the Set button to select the Enhydra server startup class, `com.lutris.multiServer.Multiserver`.

Application parameters option

If you have set the main class to launch the Enhydra server, you can use the application parameters to pass in a configuration file. For example, if you create a Web application using the Kelp Application Wizard, the application parameter is set to:

```
<source_directory>/output/conf/servlet/servlet.conf
```

Chapter 2. Setting project properties in NetBeans

After creating a NetBeans project to manage your source, you can edit Kelp project properties by selecting Project|Settings from the main menu. The project Settings window contains two Kelp-specific nodes: XML Compiler and Kelp Deployment. By selecting either node, one can set "simple" properties via the property sheet. For more robust settings, however, one should use the customizer, accessible via the node's Customize context menu option.

Node properties may be set on recognized markup language files (e.g., HTML, WML) and template files by right-clicking on the node and selecting the Customize option. The selected property can be set more quickly by using the node's property sheet, which is viewed by selecting the Properties context menu option.

Classpath

The classpath is set by mounting directories and JARs to the project's Filesystems collection. By default, some of the Enhydra JARs are mounted when NetBeans starts, and do not appear in the Filesystems tab. These JARs include:

- `<Enhydra5.1_root>/lib/enhydra.jar`
- `<Enhydra5.1_root>/lib/xmlc.jar`
- `<Enhydra5.1_root>/lib/xerces.jar`
- `<Enhydra5.1_root>/lib/tomcat.jar`
- `<Enhydra5.1_root>/lib/core.jar`
- `<Enhydra5.1_root>/lib/gnu-regexp-1.1.4.jar`
- `<Enhydra5.1_root>/lib/xhtml.jar`
- `<Enhydra5.1_root>/lib/wireless.jar`
- `<Enhydra5.1_root>/lib/build/toolbox.jar`

When using the Kelp Application Wizard, additional JARs may be added to the mounted Filesystems if they are required to build the default configuration.

Project directory

The project directory is where Kelp will look for the default input, output and source directories, or their equivalents. As NetBeans does not create a tangible project directory the way JBuilder does, the user will have to provide this directory. By default, the Project directory is assumed to be the first directory added to the project (not to be confused with the first mounted directory). This default can be overridden via the XMLC "Project Directory" property.

Source directory

Kelp uses the source directory as the parent for all generated packages. The source directory defaults to `<project_dir>/src`, but this can be overridden by adjusting the "Source Directory" XMLC property.

Output directory

Kelp for NetBeans does not have an output directory, per se. The output directory is the same as the source directory, and this is where the class files should be generated on compilation.

Deploy root

The deployment root directory can be set through the Kelp deployer property sheet.

Run options

When deploying your app, if you have chosen to generate a `StartServer.java` file, you can run your application from the IDE. In this case, a pre-configured external executor should be attached to the `StartServer` file, so you can right-click the node and select `Execute` from the context menu.

Setting up a project for use with the Kelp ApplicationWizard

If the Kelp Application Wizard is used to jumpstart your development effort, we recommend the following project setup:

- Create the project directory.
- In NetBeans, select `Project|New Project` from the main menu.
- In the `Create New Project` dialog, enter a name for your project.

Kelp does not use this project name internally, so there are no naming restrictions.

- When asked if you would like to start with a new Filesystem configuration, choose the `New` option.
- When prompted to mount a directory, select your project directory.
- Once the project is created, select the `Project` tab in the Explorer window.
- Right-click on the project root and select `Add Existing` from the context menu.
- Select your project directory and click the "OK" button.

You are now ready to generate a new application. See "Using the Kelp Application Wizard" in "Using the wizards and tools" ([html \[tools.html\]](#) , [pdf \[tools.pdf\]](#)).

Chapter 3. Setting XMLC properties

XMLC properties can be set for the project, a folder, or individual files. XMLC properties set for a file always override properties set for a folder, and folder properties always override properties set for the project.

XMLC project properties

You can view the XMLC project properties page by opening the Project|Project Properties dialog box and selecting the Enhydra XMLC tab (Figure 7). This page shows the same make and output options that you can set on the Options tab of the XML Compiler dialog.

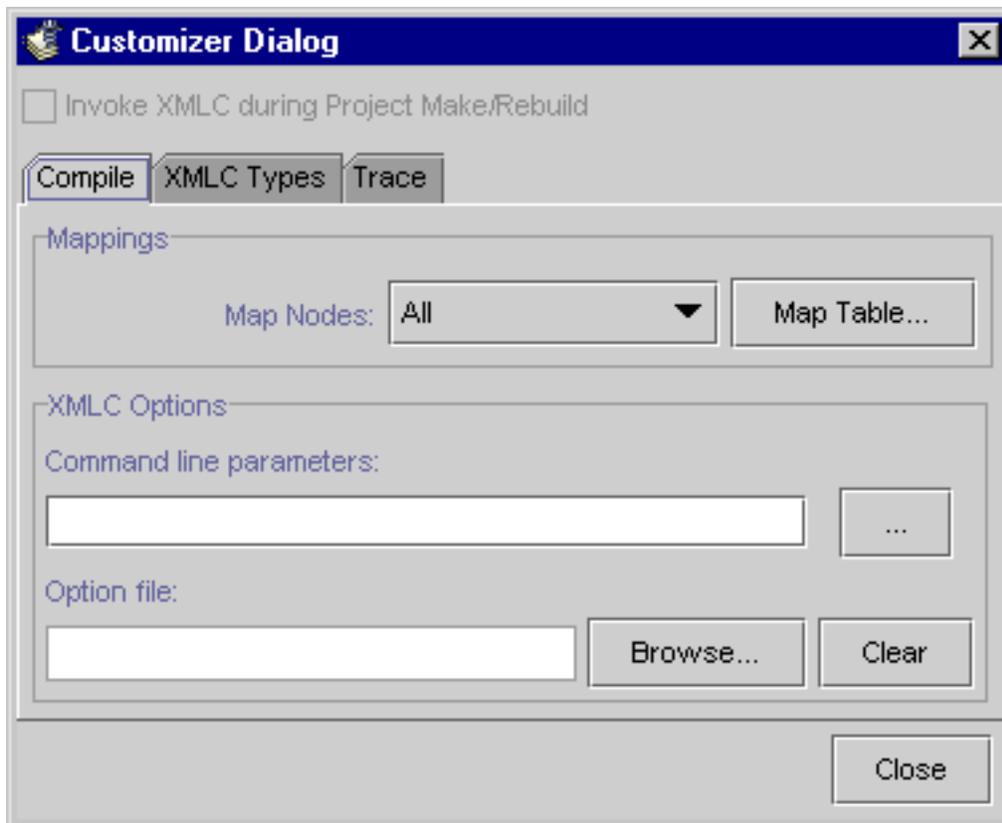


Figure 7: Project Properties page

The Compile tab includes three sections:

- Generated Java Source: Tells you which XMLC options need to be set for the current IDE.
- Mappings: Determines how class names are generated.
- XMLC Options: Lets you specify XMLC command-line options or select an XMLC Options file to set XMLC options. For more information on XMLC command-line options, see Chapter 5, "Enhydra XMLC," of the Developer's Guide.

Click Edit to open the XMLC Parameter Editor. The XMLC Parameter Editor lets you organize complex XMLC command-line options.

In the Mappings section, you can define a mapping table and set the nodes that you want to use with the table. The

mapping table lets you specify package names to use when generating DOM classes for a given directory. For example, the sample Kelp Web application project uses a mapping table to map all the HTML files in a resources directory into the `kelp.webapp.presentation` package.

To view or edit the package name mapping:

- Open the XML Compiler dialog
- Select the Options tab.
- Click the Edit button.

This opens the mapping table, where you can add a new entry that maps a directory to the correct presentation package. Here is a sample mapping for the Kelp sample project:

Source directory:

```
/user/local/jbuilder/kelp2/samples/webapp/src/kelp/webapp/resources
```

Package name:

```
kelp.webapp.presentation
```

Select the Output tab on the XMLC page in Project Properties. These settings cause the XMLC to stream out additional information when compiling the HTML files. The Output settings do not affect the generated DOM classes.

Note: For faster compiles, deselect all the Output options.

XMLC node property page

You can open the XMLC node property page by right-clicking an HTML or WML file in the Project pane and selecting Properties. This opens the Properties page for the selected file. As Figure 8 shows, this page has four options:

- Copy to Output as a Static Resource: Select if the file will be served as a static file. The file will be copied to the output directory.
- Generate DOM Class: Select if you want to select this file for compilation with XMLC.
- Generated Class Names: Lets you configure the names of generated classes. Click Map Table to open the Map editor.
- XMLC Options: Lets you enter XMLC command-line options for the selected file, or for your entire project. This option is available only when Generate DOM Class is selected.

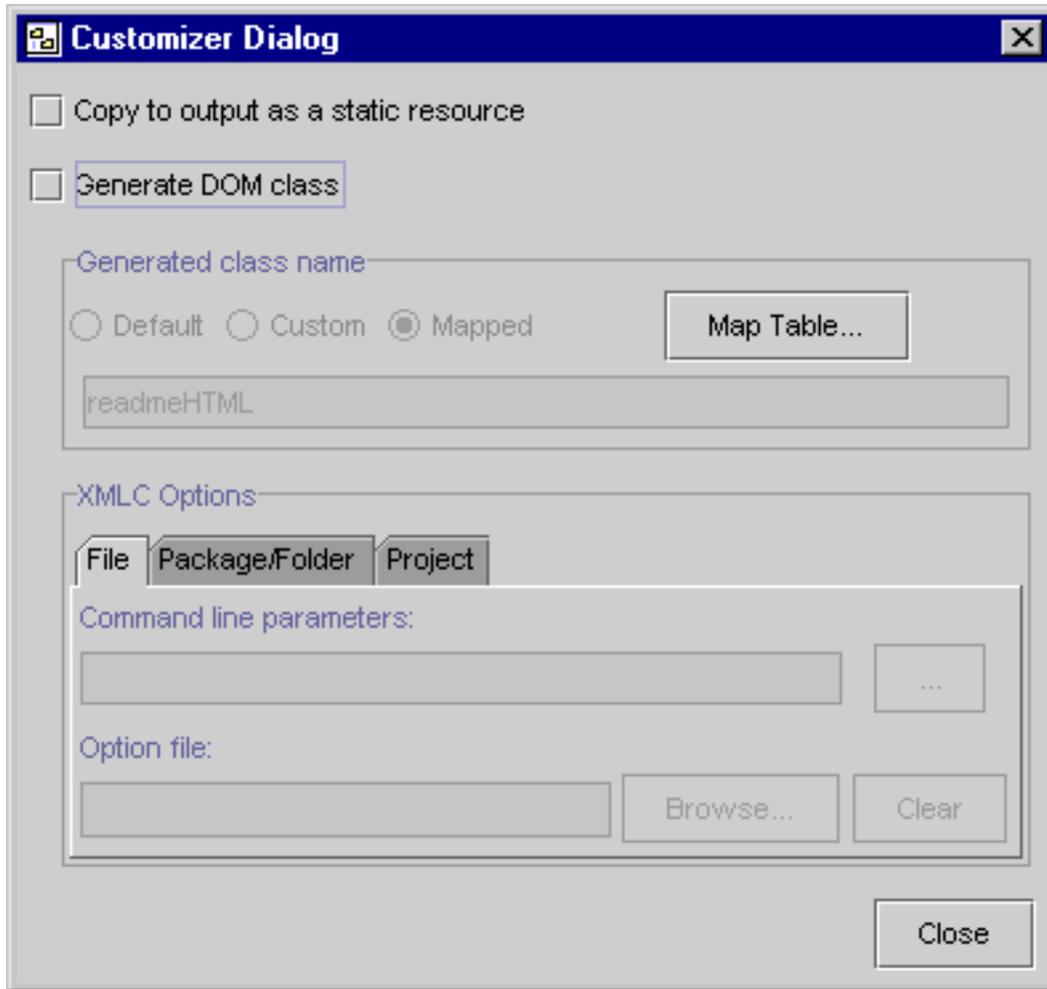


Figure 8: XMLC Node Properties page

How Kelp sets class names for XMLC

You can use the Generated Class Name section to select how you want to generate the class name. There are three choices: default, custom, and mapped. The file name, directory location, and project source path determine the default name.

For example, if you create a new application in `/home/bob/jbprojects/untitled1`, the source path for the `Welcome.html` file is `/home/bob/jbprojects/untitled1`, the directory location is `/home/bob/jbprojects/untitled1/untitled1/presentation` and the file name is `Welcome.html`. The package name is `untitled1.presentation.WelcomeHTML`.

The source path is removed from the directory location to create the new package name. The new class name is formed by removing the `.html` extension and adding "HTML". Select the Custom option if you want to enter your own class name. When using custom class naming, make sure the name you enter conforms to a valid Java identifier.

Note: If the selected HTML file is not located in a subdirectory of one of the Project source path settings, Kelp may not be able to generate a valid default class name.

To map your HTML files to user-defined class names, select Mapped and click Map Table to open the Edit Project Map dialog box. XMLC uses this mapping table to set custom package names based on the directory of the HTML

file. This mapping table is normally used to map a resources directory to a presentation package. There is only one mapping table per project.

Setting XMLC options for selected files

To set additional XMLC options for the selected file, use the XMLC Options section in the XMLC node property page (see Figure 8).

Command-line parameters let you add command-line options to XMLC from within your IDE. Click Edit to open the XMLC Parameter Editor. The Parameter Editor allows you to organize complex command-line parameters. If you have an options file for XMLC, click Set and select the options file. To clear a previously selected options file, click Clear. For more information on using XMLC, see Chapter 5, "Enhydra XMLC," of the Developer's Guide.

Input Template node property page

The Input Template node property page contains the same settings that appear on the Template tab of the Kelp Deployer project properties page. Right-click the .in file and choose Properties to open the Input Template node property page for a .in file in your project.

In addition to the project properties, there is a Generate .conf check box you can select to generate configuration files for the current template.