



The BlueJ Environment Reference Manual

Version 0.1
for BlueJ Version 1.1

Michael Kölling
School of Network Computing
Monash University

HOW DO I ... ? – INTRODUCTION	1
ABOUT THIS DOCUMENT	1
RELATED DOCUMENTS	1
1 PROJECTS	2
1.1 CREATE A NEW PROJECT	2
1.2 OPEN A PROJECT	2
1.3 FIND OUT WHAT A PROJECT DOES	2
1.4 COPY A PROJECT	2
1.5 COMPILE A PROJECT	3
1.6 DOCUMENT A PROJECT	3
1.7 EXECUTE A PROJECT	3
1.8 PRINT THE PACKAGE WINDOW	3
2 CLASSES	4
2.1 CREATE A NEW CLASS.....	4
2.2 REMOVE A CLASS	4
2.3 RENAME A CLASS	4
2.4 COMPILE A CLASS	4
2.5 EDIT A CLASS.....	4
2.6 OPEN A CLASS	4
2.7 VIEW THE INTERFACE OF A CLASS.....	4
2.8 VIEW THE IMPLEMENTATION OF A CLASS.....	4
3 PROJECT EDITING	5
3.1 CREATE AND REMOVE CLASSES.....	5
3.2 LAYOUT A PROJECT	5
3.3 MOVE A CLASS ICON	5
3.4 RESIZE A CLASS ICON	5
3.5 MOVE AN ARROW	5
3.6 ADD AN ARROW	5
3.7 REMOVE AN ARROW	5
3.8 WRITE A PROJECT DESCRIPTION.....	5
4 EDITING SOURCE CODE	6
4.1 VIEW THE SOURCE CODE OF A CLASS	6
4.2 ENTER A NEW ROUTINE.....	6
4.3 FIND ERRORS IN THE SOURCE.....	6
4.4 FIND OUT WHAT THE EDITOR CAN DO.....	6
4.5 FIND OUT WHAT A FUNCTION KEY DOES.....	6
4.6 CHANGE KEY BINDINGS	6
4.7 FIND OUT MORE	6
5 EXECUTION	7
5.1 CREATE AN OBJECT	7
5.2 CALL A ROUTINE.....	7
5.3 ENTER PARAMETERS	7
5.4 EXECUTE A STATIC METHOD	7
5.5 EXECUTE A "MAIN" METHOD	7
5.6 EXECUTE AN OBJECT THAT WAS RETURNED BY A ROUTINE.....	7
5.7 USE AN OBJECT FROM THE OBJECT BENCH AS A PARAMETER	7
5.8 STOP THE EXECUTION OF A BLUE PROGRAM.....	7
6 DEBUGGING	8
6.1 INSPECT AN OBJECT	8

6.2	SET A BREAKPOINT	8
6.3	REMOVE A BREAKPOINT	8
6.4	STEP THROUGH MY CODE	8
6.5	INSPECT VARIABLE VALUES IN MY PROGRAM.....	8
6.6	FIND OUT ABOUT THE CALL SEQUENCE AT A BREAKPOINT	8
6.7	OPEN THE DEBUGGER WINDOW.....	8
7	THE TERMINAL WINDOW	9
7.1	SHOW/HIDE THE TERMINAL WINDOW	9
7.2	CLEAR THE SCREEN OF THE TEXT TERMINAL.....	9
7.3	SAVE THE PROGRAM OUTPUT TO A FILE	9
7.4	KEEP ALL OUTPUT	9
8	CONFIGURATION	10
8.1	CHANGE BLUEJ SETTINGS	10
8.2	USE THE PREFERENCES DIALOG	10
8.3	USE A LOCAL COPY OF THE DOCUMENTATION.....	11
8.4	CONFIGURE THE WEB BROWSER	11
8.5	CHANGE FONTS.....	11
8.6	CHANGE THE DIAGRAM NOTATION STYLE	12
8.7	CHANGE THE INTERFACE LANGUAGE	12
8.8	USE A DEFAULT LOCATION FOR PROJECTS.....	13
8.9	SWITCH SYNTAX HIGHLIGHTING ON/OFF	13
8.10	USE DIFFERENT TEMPLATES FOR NEW CLASSES.....	13
9	MISCELLANEOUS.....	14
9.1	USE THE FILE SELECTION DIALOG.....	14

How Do I ... ? – Introduction

About This Document

This is the "How Do I..." reference manual for the BlueJ Application Development Environment. This manual tries to answer questions about the BlueJ environment. These questions are assumed to be of the form "How do I do this-or-that?" You can replace the *do this-or-that* part with any of the section headings, and hopefully construct a question that is both close to what you wanted to ask and answered in this manual. Of course, you can also just browse through this manual, maybe after you have used the environment for a while, and possibly find out one thing or another that you did not know about BlueJ. There are lots of tips and tricks that are not essential (you get along without them) but which are handy, neat and can speed up your work. Many of these are described in this document.

Related Documents

This document is intended as a reference manual to look up specific tasks when working with BlueJ. If you are interested in an introduction to the BlueJ system, refer to the BlueJ Tutorial.

1 Projects

1.1 Create a new project

menu: Project – New... *shortcut:* —

To create a new project, start BlueJ. Select New... from the Project menu. A file selection dialog will open. [Use the file selection dialog](#) to enter a name and location for the project.

Click Ok, and the new project is created and opened.

BlueJ will create a directory for your project, so you should be a bit careful with the name: characters that your file system cannot cope with should not be included in the project name. Also, class paths will be set automatically for your project, and Java has trouble with some characters in the class path. Generally, it is best to stick to alphanumerical characters for project names.

1.2 Open a project

menu: Project – Open... *shortcut:* Alt-o

There are two ways to open a project: from the command line of a shell (before Blue is started) type

```
bluej <project-name>
```

to start BlueJ and automatically open the project <project-name>.

If BlueJ is already running, use the Open... command from the menu. A file selection dialog will open. [Use the file selection dialog](#) to select a project to open. Click Ok, and the project is opened.

1.3 Find out what a project does

menu: *shortcut:*

A project is described in the project description. The project description is shown as a note icon close to the top left corner of the main screen:



To read the project description, double click on the note icon.

1.4 Copy a project

TBW

1.5 Compile a project

menu: Tools – Compile *shortcut:* Ctrl-k *toolbar button:* Compile

To compile a project, select Compile from the menu or toolbar. This function will make an analysis of the current project, check which classes need recompilation, check dependencies, and then compile all classes that need compilation in the appropriate order.

This function will always try to compile all uncompiled classes. If errors are detected, each class with one or more errors is opened and the first error is displayed. To compile one or more specific classes without compiling them all, see [Compile A Class](#) .

1.6 Document a project

To document a project properly, you should at least do the following: [Write a project description](#), comment all classes and comment all routines.

Class comments and routine comments are both entered in the class's source code. [Open a class](#) to enter these comments.

1.7 Execute a project

To execute a project, you first have to find out which routine you want to execute. Some projects have a hierarchical structure with one top-level class. The top-level class can have one or more start routines. Some projects have a more open structure in which you call various functions from different classes to perform the various tasks the project offers. If you do not know which class or which routine you should use have a look at the project description note. It should be described there. The project description note is further explained in [Find out what a project does](#). As a rule-of-thumb, top-level classes are usually placed in the project window near the top-left corner.

Once you know what you want to execute, you may want to [Execute a static method](#), or you may need to [Create an object](#) to start one of its methods. Once you have created that object, you can [Call a method](#).

1.8 Print the package window

menu: Project – Print... *shortcut:* Alt-p

To print a project, select Print... from the project menu.

...

2 Classes

2.1 Create a new class

2.2 Remove a class

2.3 Rename a class

2.4 Compile a class

2.5 Edit a class

2.6 Open a class

2.7 View the interface of a class

2.8 View the implementation of a class

3 Project editing

3.1 Create and remove classes

3.2 Layout a project

3.3 Move a class icon

3.4 Resize a class icon

3.5 Move an arrow

3.6 Add an arrow

3.7 Remove an arrow

3.8 Write a project description

4 Editing source code

4.1 View the source code of a class

4.2 Enter a new routine

4.3 Find errors in the source

4.4 Find out what the editor can do

4.5 Find out what a function key does

4.6 Change key bindings

4.7 Find out more

5 Execution

5.1 Create an object

5.2 Call a routine

5.3 Enter parameters

5.4 Execute a static method

5.5 Execute a "main" method

5.6 Execute an object that was returned by a routine

5.7 Use an object from the object bench as a parameter

5.8 Stop the execution of a blue program

6 Debugging

6.1 Inspect an object

6.2 Set a breakpoint

6.3 Remove a breakpoint

6.4 Step through my code

6.5 Inspect variable values in my program

6.6 Find out about the call sequence at a breakpoint

6.7 Open the debugger window

7 The terminal window

7.1 Show/hide the terminal window

menu: View – Show Terminal *shortcut:* Ctrl-t

To show or hide the Text Terminal toggle the Show Terminal item in the View menu. The text terminal is automatically popped up (shown if it was hidden or de-iconified if it was iconified) if output is written to it or input is expected.

7.2 Clear the screen of the text terminal

terminal window:

menu: Options – Clear *shortcut:* Ctrl-k

To clear the text terminal, select Clear from the Option menu in the terminal window.

The terminal can not be cleared from within a Java application.

7.3 Save the program output to a file

terminal window:

menu: Options – Save to file... *shortcut:* Ctrl-s

The contents of the text terminal can be saved to a file. This may be useful to document the output of a program. To save the terminal text, select Save to File... from the terminal's Options menu.

7.4 Keep all output

terminal window:

menu: Options – Unlimited buffering *shortcut:* —

By default, the terminal buffers only the last 48 lines of output. If you want to keep more of the output (for example to save it to a file later), switch on the Unlimited buffering option from the terminal's Options menu.

With unlimited buffering, all output is kept in the terminal. This option can make the output relatively slow if the application produces a lot of text.

8 Configuration

8.1 Change BlueJ settings

There are two ways to change configuration settings in BlueJ: User options are changed from within BlueJ. [Use the preferences dialog](#) to change those settings.

Administrator options are changed by editing the configuration file `bluej.defs`. The configuration file is at `<BLUEJ_HOME>/lib/bluej.defs` (where `<BLUEJ_HOME>` is the directory where BlueJ was installed).

The configuration file contains a list of properties in the format

```
property-name = value
```

The property names should not be changed. The values may be changed to alter the configuration. It is a good idea to make a backup of this file before you edit it.

All properties in the `bluej.defs` file apply to all users on the system. There is also a user specific configuration file for each user. It is stored in

```
<USER_HOME>/ .bluej/bluej.properties      (Unix)
```

or

```
<USER_HOME>\bluej\bluej.properties      (Windows)
```

The `<USER_HOME>` directory may vary on single user systems. Search for a file named "bluej.properties" if you are not sure where it is on your system.

Each of the properties in system configuration file (`bluej.defs`) can be specified in the user configuration file. Settings in the user configuration file override the settings in the system configuration file.

The individual properties are described in the next few sections of this manual.

8.2 Use the preferences dialog

main window:

menu: Tools – Preferences...

shortcut: —

editor:

menu: Options – Preferences...

shortcut: —

To set user preferences from within BlueJ, choose Preferences... from the menu.

The preferences you can specify here are stored in the user configuration file. (See [Change BlueJ settings](#) for more information about this file.) User preferences, if changed, will override the settings in the system configuration file.

The settings that can be changed using the preferences dialog include the editor font size, use of syntax highlighting, the Java documentation URL and the diagram notation style.

8.3 Use a local copy of the documentation

Three menu items in the BlueJ help menu display the BlueJ tutorial, the BlueJ reference manual and the Java standard class libraries. Invoking these functions opens a web browser which, by default, displays documents on a remote web site. You can change the URL to be opened to point to a local copy of these documents. This can improve efficiency and allow you, if installed on the local machine, to use the documentation while being offline. The BlueJ tutorial and reference manual can be downloaded from the BlueJ web site; the Java documentation can be downloaded from Sun at java.sun.com.

To change the URLs, edit the values for the following properties:

```
bluej.url.tutorial
bluej.url.reference
bluej.url.javaStdLib
```

If your web browser is not opened by these functions, see [Configure the web browser](#).

8.4 Configure the web browser

The command used to start your web browser is defined through the properties

```
browserCmd1
browserCmd2
winBrowserCmd1
winBrowserCmd2
```

The `winBrowserCmd` properties are used on Windows systems, the `browserCmd` properties are used on all other systems. BlueJ will first use the first property (`Cmd1`). If that fails, it will try the second one (`Cmd2`). A dollar sign (\$) in the value will be replaced by the URL to be opened. The default values for Unix are

```
browserCmd1=netscape -remote openURL($)
browserCmd2=netscape $
```

These commands first attempt to open the URL in an already running instance of Netscape and, if that fails, start a new Netscape browser. You can edit these commands if, for instance, Netscape is not in your path or you want to use a different browser.

The default values for Windows are

```
winBrowserCmd1=rundll32 url.dll,FileProtocolHandler $
winBrowserCmd2=
```

This command will attempt to start the browser that is specified as your preferred browser in the Windows settings.

8.5 Change fonts

Fonts are specified using the following properties:

```
bluej.fontsize           (font size for most interface components)
bluej.editor.fontsize    (font size for editor)
```

<code>bluej.editor.font</code>	(font face for editor)
<code>bluej.fontsize.printText</code>	(font size for printed source)
<code>bluej.fontsize.printTitle</code>	(font size for printout title)
<code>bluej.fontsize.printInfo</code>	(font size for print footer)

All font sizes are integer numbers.

The only font face that can be specified is the font used in the editor for showing class sources. Font faces are specified using a font family name followed by an optional "-bold". Examples are

```
bluej.editor.font=Monospaced
bluej.editor.font=Monospaced-bold
bluej.editor.font=SansSerif
bluej.editor.font=Arial-bold
```

The only fonts guaranteed to exist on all Java platforms are Monospaced, Serif, SansSerif and Symbol. It is generally preferable to use monospaced fonts for editing source code. Sources using a mix of TABs and spaces will not line up with other fonts.

8.6 Change the diagram notation style

BlueJ supports two different diagram notations: UML notation and Blue notation. (In versions before 1.1, only Blue notation was supported.) To set the notation style, set the value of the following property:

```
bluej.notation.style
```

The value may be either `uml` or `blue`.

This setting provides the default. Users can modify this setting [using their preferences dialog](#).

8.7 Change the interface language

The interface language can be changed using the property

```
bluej.language
```

Supported language currently include English, German and Swedish.

Language specific files are stored in `<BLUEJ_HOME>/lib/<language>` (where `<BLUEJ_HOME>` is the directory where BlueJ was installed and `<language>` is a directory with the same name as the language). The language directories contain several text files with BlueJ interface texts.

If your preferred language is not included, it is easy to make a language definition yourself. This is how:

- Copy the directory of a language you understand to a new name representing your language (e.g. copy the directory "english" to "spanish").
- Open each file within the "spanish" directory with a text editor and translate the texts.
- Set the language property in the configuration file to

```
bluej.language=spanish
```

If you make a translation for a new language for a new language, please send your translation to bluej@bluej.org - we will then include your language definition in the next release of BlueJ.

8.8 Use a default location for projects

The property

```
bluej.defaultProjectPath
```

can be used to specify the default location of BlueJ projects. The file selection dialog BlueJ uses to open projects will start in the directory specified in this property.

8.9 Switch syntax highlighting on/off

The property

```
bluej.syntaxHilighting
```

is a boolean property (its values are `true` or `false`) for switching syntax highlighting on or off. This property provides the default. Users can modify this setting [using their preferences dialog](#).

8.10 Use different templates for new classes

When creating new source files, BlueJ provides a default template for the source text. These templates are stored in `<BLUEJ_HOME>/lib/` (where `<BLUEJ_HOME>` is the directory where BlueJ was installed).

The templates and their file names are

- template for standard Java class (`stdclass.tpl`)
- template for interface (`interface.tpl`)
- template for abstract classes (`abstract.tpl`)
- template for Applet source (`applet.tpl`)
- template for JApplet source (`japplet.tpl`)
- template for applet web page (`html.tpl`)
- template for project note (`readme.tpl`)
- text added to project note when exporting (`readmeexp.tpl`)

The file `"shell.tpl"` is for internal use and should not be modified.

If you want to use a different default text for sources, you can modify these files. If you want to use different templates on a per-user-basis, different files can be created, and the template properties can be modified in the user configuration file to point to these files. The template properties are named:

```
template.*
```


9 Miscellaneous

9.1 Use the file selection dialog

TBW