JCreator “Just-In-Time” Tips

[A Reference For Jcreator Pro 2.5]

Prof. Ronald I. Frank
Pace University
September 2002

This is a work in progress. Please send corrections, comments (+ & -), and suggestions to: rfrank @ pace.edu
PREFACE

A. At Installation of JCreator: Backup The Tool Defaults

First go to page 8 and copy the <default> settings to “default backup” for each tool separately. This is because there is no “restore factory settings” for them.

B. To Begin a code:

If you start by invoking [File, New] or [Project, New], many of the class path problems never happen because the IDE requires that you put in a path to the directory containing your work. If you choose not to do this, some of the tips below help you get around problems.

C. Warning: About these Tips

1. These Tips are in no particular order or implied importance. You may not need ANY of them. They are here for you only IF you need them, when you need them.

2. In JDKTools: all 4 built in tools ALWAYS run from their own <Default>. You may have to edit <Default>.
# TABLE OF CONTENTS

**PREFACE** ....................................................................................................................................................... 2

A. At Installation of JCreator: Backup The Tool Defaults .................................................................................. 2

B. To Begin a code: ............................................................................................................................................... 2

C. Warning: About these Tips ......................................................................................................................... 2

**TABLE OF CONTENTS** ..................................................................................................................................... 3

1 - TO INSTALL THE NEW SDK PATH AND SDK DOC PATH IN JCREATOR .................................................. 7

2 - TO RESET THE DEFAULTS: .......................................................................................................................... 8

A. GENERAL Settings [Will run code w/o a project from the C: drive] ....................................................... 8

B. To Set Directories ......................................................................................................................................... 9

C. To Configure Compiler ............................................................................................................................. 10

D. To Configure Run Application .................................................................................................................. 13

E. Configure Debugger To Debug An Application ...................................................................................... 16

3 - TO RUN FROM A:\ ....................................................................................................................................... 18
A. To Define A:\ As The Working Directory [No Project Defined]

B. To Configure The Compiler To Run With A:\

C. To Configure “Run Application” From A:\

4 - TO CAPTURE OUTPUT TO THE IDE’S OUTPUT PANEL

5 - TO TELL THE IDE WHERE MAIN() IS

6 - TO SETUP FOR INPUT ARGUMENTS TO MAIN()

7 - TO SET THE IDE CODE PATH: SEPARATELY FOR EACH PROBLEM

8 - TO MAKE A PROJECT

9 - TO COMPILE AN APPLET

A. Tab Settings

B. The Compiler Run

C. The error if COMPILING when showing the .htm tab.

D. The error if trying to RUN showing the .java tab [and a no main() designated]

E. The error if trying to RUN showing the .java tab [and a main() designated]
F. The RUN Showing The .htm Tab. .......................................................................................................................... 49

10 - TO RUN AN APPLET IN A PROJECT .................................................................................................................. 50

A. The Directory Location And Contents .................................................................................................................. 50

B. Setting Up The Project ............................................................................................................................................. 51

C. Automatic Directories Setting ................................................................................................................................ 52

D. The Project Layout (code generated by the IDE) .................................................................................................. 53

E. The Java Code .......................................................................................................................................................... 54

F. The Automatically Generated .htm Code ................................................................................................................ 55

G. The Directory Location And Contents After Setup [Project And Code File] .................................................. 56

H. Compilation: [using <default>] ............................................................................................................................... 57

I. Run: The JCreator Applet Viewer Output Window [using <default>] ................................................................. 58

J. The Directory Location And Contents After Compilation and Run ......................................................................... 59

11 - TO RUN AN APPLET OUTSIDE OF A PROJECT ............................................................................................... 61

12 - TO SET “THIS DIR” (;.;) IN THE “$[CLASSPATH]” IDE VARIABLE ................................................................. 63
13 - TO SET “THIS DIR” (;;) IN YOUR WIN2K PATH ENVIRONMENTAL VARIABLE

A. Click Start .................................................................................................................................................64

B. To Append Your Path.................................................................................................................................64

14 - TO SET UP A WHITE MEDIUM OUTPUT WINDOWS 2000 OR XP WINDOW .........................66

15 - TO SETUP FONTS AND COLOR OF .JAVA TEXT SCREENS & PRINTOUTS ..........................74

16 - DEBUGGER USE .......................................................................................................................................77

Debugger Command Line’s Command List [For the output window’s prompts] ..............................80
1 - To Install the new SDK Path and SDK DOC Path in JCreator

(This assumes you have installed the DOC for the SDK. Although it is optional, you should consider it mandatory in a learning environment. I suggest installing it directly under the root director of the SDK in a directory called DOC. Later, this directory can also store other documentation such as book support files.)

If JCreator is not installed

• Install it and when asked where the SDK is, just choose the path to
  [e.g., C:\Java\j2se1.4.0]

• When asked where the DOC is, just choose the path to
  [e.g., C:\Java\j2se1.4.0\DOC\docs]

If JCreator is installed

• CLICK Configure
  o Options
    ▪ JDK Profiles (delete any that are there)
      • CLICK New
        o Select Path dialog box
          ▪ Choose (CLICK) the path to
            [e.g., C:\Java\j2se1.4.0]

    • CLICK Documentation Tab
      o Select Path dialog box
        ▪ Choose (CLICK) the path to
          [e.g., C:\Java\j2se1.4.0\DOC\docs]
2 - To Reset The Defaults:

A. GENERAL Settings [Will run code w/o a project from the C: drive]
B. To Set Directories

- Configure (on Menu bar)
- OPTIONS
  - Directories

*The bottom 2 are set on installation.*
*The top line you set to the C: directory of your work (which you have previously defined!)*
C. To Configure Compiler

- Configure (on Menu bar)
  - OPTIONS
    - JDK Tools

[Notice the Backup Default setting. Copy the <Default> settings into it.]
This enables you to easily reset the default by copying parms from the backup to the default.

- **Select Application “Compiler”**
  - Select Default
  - Click Edit

This catches compiler comments in the output window of the IDE.

[Show command line if you are having problems. Reading it can help determine the problem.]

The Parameters Tab contents Come from the right arrow button.

Choose JAVAC for -classpath and -d, then on the first list, Project Class-Path, Project Output Path, and Project Java Files. EVEN w/o A PROJECT. Notice the quotes!!!
D. To Configure Run Application

- Configure (on Menu bar)
  - OPTIONS
    - JDK Tools

[ Notice the Backup Default setting. Copy the <Default> settings into it. This enables you to easily reset the default by copying parms from the backup to the default. ]
- Select Application “Run Application”
  - Select Default
  - Click Edit

**This does NOT catch run outputs in the output window of the IDE but instead in a separate window.**

[Show command line if you are having problems. Reading it can help determine the problem.]

**Notice no main is defined!**

*The Parameters Tab contents Come from the right arrow button.*

- Choose JAVA for -classpath..
- Then on the first list, Project Class-Path, Project Output Path, and Project Java Files. EVEN w/o A PROJECT. Notice the quotes!!!
E. Configure Debugger To Debug An Application

[See section 16 - Debugger Use
[You MUST HAVE A PROJECT To Run the Debugger]
  - Configure (on Menu bar)
    - OPTIONS
      - JDK Tools
        - Debugger

[ Notice the Backup Default setting. Copy the <Default> settings into it.
This enables you to easily reset the default by copying parms from the backup to the default.

- Select “Debugger”
  - Select <Default>
- Click Edit

You will probably want to capture the debugging output in the IDE’s Output Screen. Also, You will have to indicate if you want to input args to main.
3 - To Run From A:

A. To Define A:\ As The Working Directory [No Project Defined]

Hint: If you have problems, Set the following boxes to verbose ["Show Additional .... Info"] and READ the output!
B. To Configure The Compiler To Run With A:\
   - Configure (on Menu bar)
     - OPTIONS
       - JDK Tools
- Select Application “Compiler”
  - Select Default
  - Click Edit

![JCreator JIT Tips V.3. [A Reference] © Prof. Ron Frank 2002](image)
C. To Configure “Run Application” From A:\

- Configure (on Menu bar)
  - OPTIONS
    - JDK Tools
• Select Application “Run Application”
  o Select Default
  • Click Edit

THIS SHOULD NOT HAVE TO BE RESET FOR EACH A:\ PROBLEM
4 - To Capture Output To The IDE’s Output Panel

- **Configure (on Menu bar)**
  - **OPTIONS**
    - **JDK Tools**
      - **Select Application “Run Application”**
      - **Select Default**
        - **Click Edit**

![Options dialog box showing JDK Tools configuration with a selected application](ForPDF_JCreator_TipsV3BackUp.doc)
• (Cont.) Capture Output To IDE Output Panel
  • Select Command Tab
    ○ Check Box: Check Capture Output

THIS HAS TO BE RESET TO NO CHECK MARK
WHEN YOU GO BACK TO SENDING OUTPUT
TO A SEPARATE WINDOW
5 - To Tell The IDE Where main() Is

- Configure (on Menu bar)
  - OPTIONS
    - JDK Tools
      - Select Application “Run Application”
      - Select Default
        - Click Edit

• (Cont.) Tell IDE Where main() Is (Often choosing the edit window tab is enough)
  • **Select Parameters Tab**
    o **Text Box Main(...):** Browse to find your main()
      Use the button on the right labeled “…”

**THIS HAS TO BE RESET FOR EACH PROBLEM**
Often, you can leave it blank, and click on the edit window tab for the file that contains main(). As long as this file is showing, a run will try to use its main. If you are showing a non-main() file, a run gets a main not found error.

**Code Tab, main() Contained In This File**

```java
// main() reads in N and S everything else.

public class AlgorithmDriver {
    public static void main(String[] args) {
        // read in N and S
        int N = Integer.parseInt(args[0]);
        if (N <= 1) {
            // ...
        }
    }
}
```
Code Tab, main() NOT Contained In This File - Will Cause Error
6 - To Setup For Input Arguments To `main()`

- Set parameter to read in an argument
- Configure (on Menu bar)
  - OPTIONS
    - JDK Tools
      - Select Application “Run Application”
      - Select Default
      - Click Edit
(Cont.) Set parameter to read in an argument
- Select Parameters Tab
  - Check Box: Prompt for main function arguments

**THIS HAS TO BE RESET TO NO CHECK MARK WHEN YOU GO BACK TO APPLICATIONS WITH NO ARGUMENTS**
7 - To Set The IDE Code Path: Separately FOR EACH PROBLEM

• Set Code Path
  ▪ Configure (on Menu bar)
    • OPTIONS
      □ Directories
        ▪ Top line set to your code directory
8 - To Make A Project

(You should have .java code files ready, but you are not required to have them. Good programmers always start from their own “template” of working code to save startup effort. You can start by choosing a directory and doing a File, New)

- This example has the starting path:

  C:\DATA\PACE\IS396ES02\Code\CH4\Invest4_3Proj

- This is the starting code file and directory structure

- This Is the Project, New or File, New Dialog
• This Is the Resultant Directories Entry
• After creating the project, this is the resultant Path (classes is empty)

• This is the result of adding files to the project (no change to directory contents)
• These Are the Default Compile & Run Settings
• This is a dialog that comes up when trying to run. It gives you a chance to choose settings and indicate where main() is. You don’t have to do anything for simple one-file projects.

At this point you have only the generated the built in template code showing. You CAN run this compiled project.
The Built In Template Is:
package myprojects.investment;

import java.awt.*;
import java.awt.event.*;

class Investment extends Frame {

    public Investment() {
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
                System.exit(0);
            }
        });
    }

    public static void main(String args[]) {
        System.out.println("Starting Investment... ");
        Investment mainFrame = new Investment();
        mainFrame.setSize(400, 400);
        mainFrame.setTitle("Investment");
        mainFrame.setVisible(true);
    }
}
This is the output window after setting to a white background
[see section 14 - To Set Up A White Medium Output Windows 2000 or XP Window]
This is the resulting directory structure [classes is no longer empty]
public class Investment {

    // Main method
    public static void main(String[] args) {
        System.out.println("Please input the amount to invest [double]\n\n");
        double investmentAmount = MyInput.readDouble();
        System.out.print("\nPlease input the annual interest rate [1% = .01]\n");
        System.out.println("% [double]\n\n");
        double yearRate = MyInput.readDouble();
        System.out.println("\nPlease input the number of years [integer]\n\n");
        int years = MyInput.readInt();
        System.out.println("\nThe future value is: $" + futureInvestmentValue(investmentAmount, yearRate, years)+"\n");
    } // end main()

    // The investment function
    // It has to be static to be called by a static main()
    public static double futureInvestmentValue(double investmentAmount, double yearRate, int years) {
        return investmentAmount*Math.pow((1+(yearRate/12.)),(years*12));
    } // end futureInvestmentValue()
9 - To Compile An Applet

(I have “Capture Output” and “Show Command Line” checked on the Command Tab in JDK Tools in Configure Options)

A. Tab Settings

- **Compile** the .java code with the **code tab showing** otherwise, if you are showing the .htm tab, you will get an error.

- **Run** the .java code with **.htm tab showing** otherwise, if you are showing the .java tab, you will get an error.
B. The Compiler Run

```java
/* This is an example of Java applets */
import java.awt.Graphics;

public class NoProjApplet extends java.applet.Applet
{
    public void paint(Graphics g)
    {
        g.drawString(" NoProjApplet!", 10, 10);
    }
}
```

Configuration: JDK version 1.3 (Default):

Process completed.
C. The error if COMPILING when showing the .htm tab.

```
--------------------Configuration: JDK version 1.3 <Default>--------------------
javac: invalid argument: C:\DATA\PACE\IS396ES02\Code\CH2\NoProjectApplet\NoProjApplet.html
Usage: javac <options> <source files>
where possible options include:
  -g                        Generate all debugging info
  -g:none                   Generate no debugging info
  -g:{lines,vars,source}    Generate only some debugging info
  -O                        Optimize; may hinder debugging or enlarge class file
  -nowarn                   Generate no warnings
  -verbose                  Output messages about what the compiler is doing
  -deprecation              Output source locations where deprecated APIs are used
  -classpath <path>         Specify where to find user class files
  -sourcepath <path>        Specify where to find input source files
  -bootclasspath <path>     Override location of bootstrap class files
  -extdirs <dirs>           Override location of installed extensions
  -d <directory>            Specify where to place generated class files
  -encoding <encoding>      Specify character encoding used by source files
  -target <release>         Generate class files for specific VM version
```

D. The error if trying to RUN showing the .java tab [and a no main() designated].

```
--------------------Configuration: JDK version 1.3 <Default>--------------------
Command   : "C:\Java\jdk1.3\bin\java.exe" -classpath "C:\DATA\PACE\IS396ES02\Code\CH2\NoProjectApplet;C:\Java\jdk1.3\jre\lib\rt.jar;C:\Java\jdk1.3\jre\lib\i18n.jar;C:\Java\jdk1.3\lib\dt.jar;C:\Java\jdk1.3\lib\tools.jar" NoProjApplet
Directory : C:\DATA\PACE\IS396ES02\Code\CH2\NoProjectApplet
java.lang.NoSuchMethodError: main
Exception in thread "main"
Process completed.
```

E. The error if trying to RUN showing the .java tab [and a main() designated].

```
--------------------Configuration: JDK version 1.3 <Default>--------------------
Command   : "C:\Java\jdk1.3\bin\java.exe" -classpath "C:\DATA\PACE\IS396ES02\Code\CH2\NoProjectApplet;C:\Java\jdk1.3\jre\lib\rt.jar;C:\Java\jdk1.3\jre\lib\i18n.jar;C:\Java\jdk1.3\lib\dt.jar;C:\Java\jdk1.3\lib\tools.jar" NoProjApplet
C:\DATA\PACE\IS396ES02\Code\CH4\RecursionArgs\RecursionArgs.class [This is what I had in my main() designation RIF]
Directory : C:\DATA\PACE\IS396ES02\Code\CH2\NoProjectApplet
java.lang.NoSuchMethodError: main
Exception in thread "main"
```
F. The RUN Showing The Applet Viewer Window And The .htm Tab.
10 - To Run An Applet In A Project

A. The Directory Location And Contents

[Initial Code Root Directory everything empty]
B. Setting Up The Project
C. Automatic Directories Setting

![Image of JCreator Options window showing directories settings]

- Default Project Directory: C:\DATA\PACE\1S396E\SC2\Code\CH2\HelloApplet2\...
- Syntax Directory: C:\Program Files\JCreator\Creator LE\Syntax\...
- Project Templates Directory: C:\Program Files\JCreator\Creator LE\Templates\...
D. The Project Layout (code generated by the IDE)

```java
5 * directory ..\JCreator\Templates\Temporary
6 *
7 * You can also create your own project template by
8 * folder in the directory ..\JCreator\Template\.
9 * templates as examples.
10 *
11 */
12
13 import java.awt.*;
14 import java.applet.*;
15
16 public class AppletProject extends Applet {
17
```

Shows the help document of the selected text
E. The Java Code

```java
/*
 * @(#)AppletProject.java 1.0 02/03/24
 *
 * You can modify the template of this file in the directory ..\JCreator\Templates\Template 2\Project Name.java
 *
 * You can also create your own project template by making a new folder in the directory ..\JCreator\Template\. Use the other templates as examples.
 *
 */

import java.awt.*;
import java.applet.*;

public class AppletProject extends Applet {

    public void init() {
    }

    public void paint(Graphics g) {
        g.drawString("Welcome to Java!", 50, 60);
    }
}
```
F. The Automatically Generated .htm Code

```html
<HTML>
<HEAD>
</HEAD>
<BODY bgcolor="#000000">
<CENTER>
<APPLET
code = "AppletProject.class"
width = "500"
height = "300"
>
</APPLET>
</CENTER>
</BODY>
</HTML>
```
G. The Directory Location And Contents After Setup [Project And Code File]
H. Compilation: [using <default>]

```java
/* @(#)AppletProject.java 1.0 02/
 * You can modify the template of directory ..\JCreator\Template

--------------Configuration: AppletProject - JDK version 1.3
Command       : "C:\Java\jdk1.3\bin\javac.exe" -classpath 'C:\DATA\PACE'
Directory      : C:\DATA\PACE\IS396ES02\Code\CH2\HelloApplet2\AppletProje
Process completed.
```

For Help, press F1  Ln1, Col1, Char1  DOS  NUM
I. Run: The JCreator Applet Viewer Output Window [using <default>]

( running with the either the .htm or the .java file showing )

[Window was set to a white background to save printing ink.]

[See Section 15 - To Setup Fonts and Color of .java Text Screens & Printouts]
J. The Directory Location And Contents After Compilation and Run

[Project And Code File]

HINT: Notice if the bar is gray except the lightening button, which is red, and nothing can work. You left the output window running! Click the lightening button to cancel it.
11 - To Run An Applet Outside Of A Project

- Configure (on Menu bar)
  - OPTIONS
    - JDK Tools
      - Select Application “Run Applet”
        - Select Default
          - Click Edit
(Cont.) Run An Applet

- Command Tab & Parameters Tab

![Image of JCreator JCreator JIT Tips V.3. [A Reference] © Prof. Ron Frank 2002](image-url)
12 - To Set "This Dir" (;.;) In The "[$Classpath]" IDE Variable

- In Either Compiler or Run Application
  - Type ;.; onto "[$Classpath]" -> "[$Classpath] ;.;"
13 - To Set “This Dir” (;.;) In Your WIN2K Path Environmental Variable

**SAVE A COPY OF THE PATH IN A NOTEPAD DOC BEFORE STARTING**

A. Click Start

- Settings
  - Control Panel
    - 2xClick SYSTEM
      - Click System Properties ADVANCED Tab
        - Choose Environment Variables … Bar
          - If there is a User Variable “CLASSPATH”
            - APPEND AT THE END “;.;” if it does not exist.
          - OR
          - If there is NO User Variable “CLASSPATH”
            - Ignore it.
          - If there is a Systems Variable “Path”
            - APPEND AT THE END “;.;” if it does not exist.

B. To Append Your Path

- Select “Path”
  - Click Edit Button
    - Edit the Variable Value To Include Your Path
      - Click OK, Click OK, Click OK
14 - To Set Up A White Medium Output Windows 2000 or XP Window

- To Get A Window To Show: Configure, Options, JDKTools, Run Application:
  - Don’t Choose “Close Console On Exit”
  - Don’t Choose “Capture Output”
- To Set Windows Properties (When It Finally Shows Up When Running Code)
  - In The OUTPUT Window
    1. Click Upper Left Hand Corner Icon
    2. Choose Properties

- Set Screen Background To White (This saves ink when printing)
• Set Screen Text To Black
• Set Window Size Width To 50 And Height To 43 [Or What Ever You Need]
- Set Screen Buffer Size Width And Height To Whatever Your Output Needs
  This might be very large in height if you put out many pages of output.

- Set Output Screen Fonts [Any You Want] This Way
• Set Window Size To Partial Screen
When Exiting ON “OK”
• Set Properties For ALL Windows Of This “Same Title” [i.e., for many runs]
15 - To Setup Fonts and Color of .java Text Screens & Printouts

- Configure, Options, Documents, Java
  - Fonts
    - Choose Courier New (a fixed width font) size 12
    - Choose Script Western, Both (Screen & Printer)
    - Click Apply
• Colors
  • Click a Category and choose Bold or Highlight line
    Don’t’ touch Italics
  • Repeat for ALL categories 1-AT-A-TIME
  • Click Apply

• Configure, Options, Documents, Plain [Do the same as Java above]
16 - Debugger Use
[You MUST HAVE A PROJECT To Run the Debugger]
[See section 2- E. Configure Debugger To Debug An Application]

- **Configure, Options, JDKTools, Compiler, <Default>, Edit, Parameters Tab**
  - Check "Include debug info box"
  - **Compile again.**
• Configure, Options, JDKTools, Debugger, <Default>, Edit
  ▪ Command Tab: Check “Capture Output” if you want it in the IDE.
  
  ![Tool Configuration: Debugger screenshot]

  ▪ Application Parameters Tab: Check “Prompt for main method arguments” if you need it.
Debugger Command Line’s Command List [For the output window’s prompts]

Meta Instructions
Startup commands can be placed in either "jdb.ini" or ".jdbrc" in user.home or user.dir

Definitions

----  No argument
[ ]  Regular Expression Optional argument
< >  Regular Expression Required argument
|  Regular expression “or”
<class id>  Full class name with package qualifiers or a pattern
            with a leading or trailing wildcard ("*").
<expr>  A Java(tm) Programming Language expression.
         Most common syntax is supported.
<thread id>  Thread number as reported in the 'threads' command

Startup
run  [class [args]]  Start execution of application's main class

Class, Method,& Variable Info
class  <class id>  Show details of named class
classes  ----  List currently known classes
dump  <expr>  Print all object information
eval  <expr>  Evaluate expression (same as print)
exclude  [class id | "none"]  Do not report step or method events for specified classes
fields  <class id>  List a class's fields
list  [line # | method]  Print source code
locals  ----  Print all local variables in current stack frame
lock  <expr>  Print lock info for an object
methods   <class id>   List a class's methods
print     <expr>   Print value of expression
redefine  <class id> <class file name>   Redefine the code for a class
set       <lvalue> = <expr>   Assign new value to field/variable/array element

**Threads & Stack**

down   [n frames]   move down a thread's stack
interrupt   <thread>   Interrupt a thread
kill   <thread> <expr>   Kill a thread with the given exception object
pop     ----   Pop the stack through and including the current frame
reenter   ----   Same as pop, but current frame is reentered
resume   [thread id(s)]   Resume threads (default: all)
suspend   [thread id(s)]   Suspend threads (default: all)
thread   <thread id>   Set default thread
threads   [threadgroup]   List threads
threadgroup   <name>   Set current threadgroup
threadgroups   ----   List threadgroups
threadlocks   [thread id]   Print lock info for a thread
up   [n frames]   Move up a thread's stack
where   [thread id] | all   Dump a thread's stack
wherei   [thread id] | all   Dump a thread's stack, with pc info

**Execution Control**
catch   <class id>   Break when specified exception thrown
classpath   ----   Print classpath info from target VM
clear   ----   List breakpoints
clear   <class id>::<line>   Clear a breakpoint at a line
clear   <class id>,<method>[(argument type,...)]   Clear a breakpoint in a method
cont        ----  Continue execution from breakpoint
ignore  <class id>  Cancel 'catch' for the specified exception
monitor  <command>  Execute command each time the program stops
monitor  ----  List monitors
unmonitor  <monitor#>  Delete a monitor
next       ----  Step one line (step OVER calls)
step       ----  Execute current line
step up    ----  Execute until the current method returns to its caller
stepi      ----  Execute current instruction
stop at    <class id>::<line>  Set a breakpoint at a line
stop in    <class id>,<method>[(argument type,...)]  Set a breakpoint in a method
trace methods [thread]  Trace method entry and exit
untrace methods [thread]  Stop tracing method entry and exit
use ([sourcepath) [source file path]  Display or change the source path
watch [access | all] <class id>,<field name>  Watch access/modifications to a field
unwatch [access | all] <class id>,<field name>  Discontinue watching access/modifications to a field

Garbage Collection
disablegc  <expr>  Prevent garbage collection of an object
enablegc  <expr>  Permit garbage collection of an object

Command Line Control & Utility
!!  Repeat last command
----  <n> <command>  Repeat command n times
help (| ?)  ----  List commands
exit (| quit)  ----  Exit debugger
read  <filename>  Read and execute a command file
version  ----  Print version information