TABLE OF CONTENTS (HIDE)

Programmer's Survival Guide for Windows

CMD Shell, File System & Source-code Editors

1. CMD Command Prompt

Programmers use command-line to issue commands to the operating system, instead of via Graphical User Interface (GUI). This is because command-line is much more flexible than graphical interface - you can provide additional options easily to a command; you can pipe the output of one command into another command; you can put a set of commands in a script and run the script to automate a task.

The *Command Shell* (also called *Command Interpreter*, *Command Prompt*, *DOS Prompt*) is a *text-based interface* used for issuing commands to interact with the Windows Operating System. It provides a set of commands and utilities. It also has its own programming language for writing batch files (shell scripts).

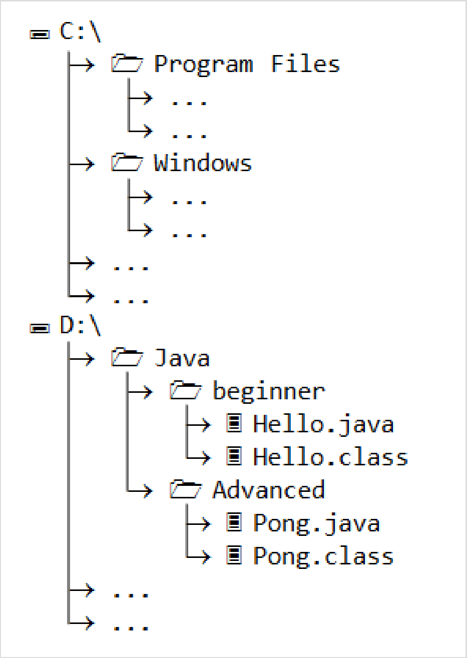
There are two versions of command shell: the newer "`cmd.exe`" (since Windows 2000), and the older and obsolete "`command.com`" (since the legacy DOS). There are kept in directory "`Windows/System32`".

You can launch a CMD shell via:

1. "Start" button ⇒ "Run..." ⇒ Enter "cmd", or
2. "Start" button ⇒ All Programs ⇒ Accessories ⇒ Command Prompt.

The CMD shell displays a prompt, in the form of "`*drive*:\*current-directory*>`" (ends with "`>`"). You can enter your command after the prompt. In this article, I shall denote the command prompt simply as "`>`".

2. Windows' File System and Basic File/Directory Commands



In Windows, files are organized in *directories* (aka *folders*). The directories are organized in a hierarchical *tree* structure, starting from the so-called *root* directory (as illustrated). A directory may contain sub-directories and files. A sub-directory may contain sub-sub-directories and files. The root directory is denoted as "`\`" (back-slash).

Windows' file system is further organized in drives, identified by a drive letter followed by a colon, e.g., `C:`, `D:` and `E:`. Each drive has its own root directory, such as `C:\`, `D:\` and `E:\`.

Windows' file system is NOT case-sensitive, a *rose* is a *Rose*, and is a *ROSE*.

2.1 Drive, Pathname and Filename

To reference a file in Windows' file system, you need to provide the *drive letter*, the *directories' name* (aka *pathname*) and the *filename*. For example, in "`C:\Program Files\java\jdk1.7.0_07\bin\javac.exe`", the drive is `C:`, the pathname is "`\Program Files\java\jdk1.7.0_07\bin`" and the filename is "`javac.exe`". The leading "`\`" (back-slash) denotes the root directory. The sub-directories are also separated by "`\`" (back-slash).

The pathname can be specified in two ways:

1. Absolute Pathname: An absolute pathname begins from the root directory of a drive. It starts with `X:\`, where `X` denotes the drive letter and the leading "`\`" denotes the root, and contains all the sub-directories leading to the file separated by "`\`", e.g., "`C:\Program Files\java\jdk1.7.0_07\bin`".

2. Relative Pathname: A relative pathname is relative to the so-called *current drive* and *current working directory*. For example, if the current drive and working directory is "`C:\Program Files\java`", then the relative path "`jdk1.7.0_07\bin`" refers to "`C:\Program Files\java\jdk1.7.0_07\bin`". A relative pathname does NOT begin with a leading "`\`" (back-slash).

2.2 Current Drive and Current Working Directory

Each CMD session maintains a so-called *current drive* and *current working directory*, which is shown in the prompt in the form of "*`drive`*`:\`*`current-directory`*`>`". All relative pathnames/filenames are relative to this current drive and working directory.

2.3 Set Current Drive (`x:`) Command

To set or change the current drive, enter the drive letter followed by a colon (`:`), e.g.,

```
Prompt> d:    // Change the current drive to D
D:\...> c:    // Change the current drive to C
C:\...>
```

2.4 Change Directory (`cd`) Command

To change current working directory, *under the current drive*, use command "`cd` *`new-path`*" (change directory).

It is important to take note that you need to set the current drive first (via "`x:`" command) before setting the current directory under the current drive.

You can specify *new-path* in two ways: *absolute* or *relative*. An absolute path begins with a "`\`" or root directory. A relative path is relative to the current working directory and does NOT begin with a leading "`\`". For example,

```
Prompt> c:
   // Set current drive to C
C:\....> cd \
   // Set current directory to the root directory of the current drive
C:\> cd Windows
   // Set current directory to "Windows" relative to current directory of the current drive
C:\Windows> cd system
   // Set current directory to "system" relative to current directory of the current drive
C:\Windows\system> cd \myproject\java
   // Set current directory absolutely to "\myproject\java" of the current drive
C:\myproject\java> cd "\Program Files\java\jdk1.7.0_07\bin"
   // Set current directory absolutely. Enclosed with double quotes if pathname contains blank.
C:\Program Files\java\jdk1.7.0_07\bin> d:
   // Set the current drive to D drive
D:\....> cd \
   // Change directory to the root of the current drive
D:\> cd Java
   // Change directory to the "Java" sub-directory of the current directory
D:\Java>
```

Take note that:

1. You need to set the *current drive* and *current directory* in *two* commands: `X:` and `cd`.
2. The current drive and current working directory is displayed in the command prompt before the "`>`".

You can `cd` in multiple stages (e.g., one `cd` for each sub-directory - recommended), or `cd` in one single stage with the full pathname.

```
Prompt> c:                               // C:\...
C:\....> cd \                            // C:\
C:\> cd Program Files                    // C:\Program Files
C:\Program Files> cd java                // C:\Program Files\java
C:\Program Files\java> cd jdk1.7.0_07    // C:\Program Files\java\jdk1.7.0_07
C:\Program Files\java\jdk1.7.0_07> cd bin   // C:\Program Files\java\jdk1.7.0_07\bin
C:\Program Files\java\jdk1.7.0_07\bin>

// Same As
Prompt> c:
C:\....> cd \Program Files\java\jdk1.7.0_07\bin
C:\Program Files\java\jdk1.7.0_07\bin>
```

You can use "`..`" (double-dot) to refer to the parent directory and "`.`" (single-dot) to refer to current directory. For example,

```
C:\Program Files\java\jdk1.7.0_07\bin> cd ..   // Parent directory
C:\Program Files\java\jdk1.7.0_07> cd ..
C:\Program Files\java> cd ..
C:\Program Files>
```

Setting proper working directory is important. For example, to compile a Java program called "`Hello.java`" stored in "`D:\myproject\java`":

1. Set the working directory to "`D:\myproject\java`", and reference the file relatively with filename only (without the path):

```
Prompt> d:
D:\...> cd \myproject\java
D:\myproject\java> javac Hello.java    // Filename only, in current directory
```

2. You can also refer to a file with its full path in any working directory:

```
// Any working directory
Prompt> javac d:\myproject\java\Hello.java
```

Although setting working directory is NOT really a necessity, it can be very messy at times without the correct working directory (e.g., the output file "`Hello.class`" is created in the current directory by default). You should always set a proper working directory before issuing any command with file references!

2.5 Directory (`dir`) Command

You can list the contents of the current directory via the `dir` command, for example,

```
Prompt> dir                // List of contents of the current directory
......
Prompt> dir Hello.java     // Show the file "Hello.java" only
```

Wildcards `*` and `?`

You can use wildcards for pattern matching. The wildcard `*` matches zero or more (any) characters; `?` matches one (any) character.

```
Prompt> dir *.java  // List files ending with ".java"
.....
Prompt> dir test*   // List files starting with "test"
.....
```

You could, of course, view the contents of a directory using "Computer" or "Window Explorer" more conveniently.

2.6 Shortcut Keys in CMD Shell - Important

Previous Commands (IMPORTANT): You can use the up/down arrow keys to scroll through the previous/next command in the command history.

Copy/Paste: You need to enable Copy/Paste by clicking on the CMD icon (top-left corner) ⇒ Properties ⇒ Options ⇒ Edit Options ⇒ Check "QuickEdit Mode". Once enabled, you can right-click to copy the highlighted text, and another right-click to paste on the command-line.

2.7 Other Basic Commands

```
Prompt> help                 // Show all the available commands
Prompt> help command-name    // Show the help menu for a specific command, e.g., "help cd"

Prompt> cls                  // Clear screen
Prompt> type filename        // Display (Type-out) the file on screen
```

2.8 Tips and Tweaks

1. CMD shell is NOT case-sensitive.
2. The screen buffer size (controlling the amount of messages retained in the screen) can be configured under "Properties" ⇒ "Layout". You should set to a bigger number (500-2000), so that you can view more old messages.

3. Keyboard Shortcuts (for Windows)

Good programmers uses keyboard shortcuts instead of mouse to perform most of the editing tasks, such as positioning the cursor, selecting texts, copy and paste. Below are the frequently-used keyboard shortcuts for the programmers.

Keyboard Shortcut	Function
ctrl+c, ctrl+v, ctrl+x	Copy, Paste, Cut
ctrl+s	Save
ctrl+f	Find
ctrl+z, ctrl+y	Undo, Redo
ctrl+RightArrow, ctrl+LeftArrow	Goto next/previous word
Home, End	Goto begin/end of the current line

ctrl+Home, ctrl+End	Goto top/end of document
ctrl+a	Select all
ctrl+shift+RightArrow, ctrl+shift+LeftArrow	Select words
shift+DownArrow, shift+UpArrow	Select lines
shift+RightArrow, shift+LeftArrow	Select characters
shift+End, shift+Home	Select till end/begin of current line
alt+Tab	Switch between open applications
alt+F4	Close the current application

Mouse Clicks

1. Single-click to position the mouse pointer.
2. Double-click to select a word.
3. Triple-click to select a paragraph.

4. Source-Code Editors (for Windows)

A *program editor* (or *source code editor*) is *programming language sensitive* and *context-aware*. It highlights the syntax elements of your programs; and provides many features that aid in your program development (such as auto-complete, compile/build/run, help menu, etc.). On the other hand, a *plain text editor*, such as "NotePad", is not language-sensitive and, therefore, is NOT suitable for writing programs. For full-scale software development, you should use an appropriate IDE (Integrated Developement Environment).

It is important to use a *mono-space font* (such as "Courier New", "Consola") for programming, so that the columns are properly aligned.

There are tons of free *program editors* available. Here are some program editors that I am currently using. Unfortunately, there isn't one magic wane (or silver bullet) that suits all my programming needs.

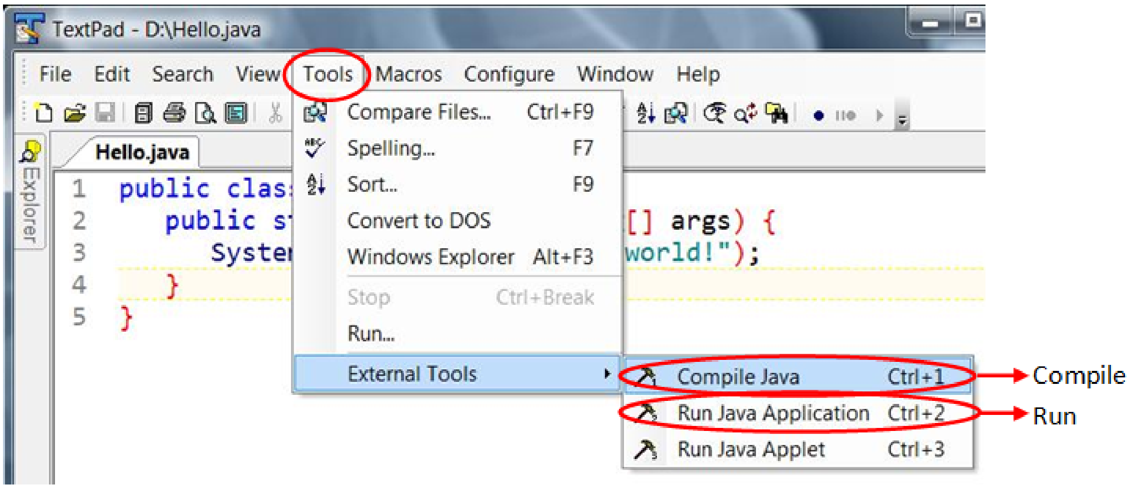
4.1 TextPad

TextPad is an excellent source-code editor, especially for writing *toy* Java programs. It is simple to use, fast to launch, and it can be configured to run JDK commands directly. TextPad is a shareware (not a freeware). The *full-function* version can be downloaded from www.textpad.com. It has no time limit but a message will keep reminding you to purchase the shareware.

To install, simply run the downloaded installer.

Using TextPad for Writing Toy Java Programs

TextPad is great in writing *toy* Java programs, as it can be configured to run JDK command directly, bypassing the command shell. From the TextPad editor, you can invoke the JDK compiler and runtime directly via "Tools" ⇒ (Version 5 only) "External Tools" ⇒ "Compile Java" or "Run Java Application". Take note of the keyboard shortcuts.



```
public clas
   public s                [] args) {
      Syste                world!");
   }
}
```

If you cannot find these commands in the "Tools" menu, goto "Configure" ⇒ Preferences... ⇒ Tools ⇒ Add ⇒ JDK Commands. You can also configure the attributes of "compile" and "run" there, such as prompting for command-line arguments.

Tips for Using TextPad

1. **Error Message Hyperlink:** Double-click on the first-line of an error message will hyperlink to the corresponding source statement.

2. **Find & Replace:** Most important commands for programmers (after the copy/paste), available under the "Search" menu.
3. **Block Select Mode:** The Block Select mode allows you to operate on a block of texts, such as deleting a few columns and filling in blank columns. To enable/disable Block Select mode, select "Configure" menu ⇒ check/uncheck "Block Select Mode".
4. **Line Numbers:** To display the line numbers, choose "View" menu ⇒ check the "Line Number" option.
5. **Command-Line Arguments:** To provide command-line arguments to your Java program in TextPad, choose "Configure" menu ⇒ "Tools" ⇒ "Run Java Application" ⇒ check "Prompt for parameters" box. When running your program, place your command-line arguments after `$BaseName` (which denotes the Java program name).

Configuring TextPad

TextPad is highly configurable to suit your personal programming style (such as your preferred font, color, layout, tab spaces, etc). Select "Configure" menu ⇒ Preferences.

1. **Font:** "Configure" ⇒ "Preferences" ⇒ Expand "Document classes" node ⇒ Select your desired class, e.g., "Java", and expand the node ⇒ "Font" ⇒ Select *monospace* font for programming, e.g., Courier New, Courier, Consola, Lucida Console.
2. **Add File Association (e.g., *.xml to use HTML Syntax Highlighting):** "Configure" ⇒ "Preferences" ⇒ Expand "Document classes" node ⇒ Click on your desired class, e.g., "HTML" ⇒ In "Files in class xxx", add your file extension, e.g., `*.xml`, `*.jsp`.

TextPad Common Errors

```
ERROR MESSAGE: error: illegal character: \29
SOLUTION:
  You has non-printable character "\29" (Ctrl-]) in your source code. (You probably meant to use Shift-] for '}'.)
  Remove this character if you can figure out its location. Otherwise, remove the whole line and re-enter the line.
```

4.2 NotePad++

NotePad++ is an open-source and free program editor that supports syntax highlighting for many languages. It is meant as a "Notepad replacement". You can download NotePad++ from http://notepad-plus-plus.org.

Plug-ins

Notepad++ is highly expandable via third-party plug-ins. Google "notepad++ plug-in" to search for plug-in. To install a plug-in, download, unzip, and move the "`dll`" file into notepad++'s "`plugins`" directory.

These are the useful plug-ins:

1. **Hex Editor Plug-in:** The hex editor plug-in allows you to view/edit binary files in hex codes. Google "notepad++ plugin hex editor", or from http://sourceforge.net/projects/npp-plugins/files ⇒ from the file list, select "Hex Editor" ⇒ download ⇒ unzip ⇒ move the `dll` file into notepad++'s "`plugins`" directory. The Hex Editor will be available from the "Plugins" menu ⇒ "Hex Editor", or the "H" icon on the menu bar.

Tips on Using NotePad++

1. **Reformatting Source Code:** TextFX ⇒ TextFXEdit ⇒ Reindent C++ Code (which also work for java codes).
2. **Block Select:** You can select columns of text by holding down the alt key while selecting (similar to WinWord).
3. **Converting Source Code to HTML:** TextFX ⇒ TextFXConvert ⇒ Encode HTML (`&<>"`).
4. **Converting End-of-line (EOL) between Windows, Mac and Unix:** Edit ⇒ EOL Conversion ⇒ Choose Windows, Unix or Mac format.
5. **Tile 2 Winodws:** Right-click on a window ⇒ Move to other view.

NotePad++ for Java Programming

To compile and run Java programs with a single hot-key (just like TextPad), I wrote two batch files called "`JavaCompile.bat`" and "`JavaRun.bat`" and save them in the NotePad++ installed directory (or a directory in the `PATH`).

JavaCompile.bat

```
@REM To run with: JavaCompile $(CURRENT_DIRECTORY) $(FILE_NAME)
@cd /d %1
javac %2
@pause
```

JavaRun.bat

```
@REM To run with: JavaRun $(CURRENT_DIRECTORY) $(FILE_PART)
@cd /d %1
java %2
@pause
```

To install the batch files, select the "Run" menu ⇒ "Run...", enter "`JavaCompile $(CURRENT_DIRECTORY) $(FILE_NAME)`" ⇒ Choose "Ctrl-1" as the hot-key for compiling Java program. Similarly, select the "Run" menu ⇒ "Run...", enter "`JavaRun $(CURRENT_DIRECTORY) $(NAME_PART)`" ⇒ Choose "Ctrl-2" as the hot-key for running Java program. (To use Ctrl-1, Ctrl-2, you need to re-configure and remove the existing mappings for these keys via "Settings" ⇒ "Shortcut Mapper...".)

These are the notepad++ internal variables:

- `$(FULL_CURRENT_PATH)`: the fully qualified path and name to the current document.
- `$(CURRENT_DIRECTORY)`: The directory the current document resides in.
- `$(FILE_NAME)`: The filename of the document, without the directory.
- `$(NAME_PART)`: The filename without the extension.
- `$(EXT_PART)`: The extension of the current document.

To modify the commands, you need to edit "`shortcuts.xml`" (search under your user's `AppDate` directory).

The above Compile-Java batch file, unfortunately, does not automatically save the current file before running the compilation. To automatically save the file, I have to use the NppExec Plugin, as follows:

1. Goto "Plugins" menu ⇒ Select "NppExec". (If you cannot find NppExec under plugins, you need to first install the NppExec plugin.)
2. Choose "Execute..." and enter the following commands:

   ```
   NPP_SAVE
   cd $(CURRENT_DIRECTORY)
   javac $(FILE_NAME)
   ```

   Save it as script name "Java-Compile".
3. Press "OK" to run the script. You shall see the output on the NppExec's console.
4. To install a hot-key for this NppExec script:
   a. From "Plugins" menu ⇒ Select "NppExec" ⇒ "Advanced Options...".
   b. In "Menu Item" ⇒ "Associated script" ⇒ Select the script "Java-Compile" ⇒ "Add/Modify". You shall see the item appears in the "Menu Items". If you open the NppExec menu, you will see "Java-Compile" as a menu item.
   c. To install a hot-key (such as Ctrl-F1 - make sure it is not used) ⇒ Goto "Settings" ⇒ "Shortcut Mapper" ⇒ "Plugin Commands" ⇒ Select "Java-Compile" ⇒ Modify ⇒ Assign your chosen hot-key.

The above "JavaRun" batch file is fine, as there is no need to save the current file.

NotePad++ for C/C++ Programming

Similarly, these scripts are used for compiling toy C/C++ programs:

GccCompile.bat

```
@cd %1
g++-4 %2.cpp -o %2
@pause
```

GccRun.bat

```
@cd %1
%2
@pause
```

4.3 Other Program Editors

PSPad

PSPad is an excellent program editor, especially for web programming such as writing JavaScript. It is a *freeware* and can be download from www.pspad.com. To install, simply run the downloaded installer. The "Help" menu provides many good links to various online programming guides and references.

Editra

Editra (@ http://editra.org): for Windows, Mac and Linux

4.4 Hex Editors

At times, you may need to display the hex values of a file. A *Hex Editor* is a handy tool that a good programmer should have in his/her toolbox. There are many freeware/shareware Hex Editor available. Try google "Hex Editor".

I recommend notepad++ with the "Hex Editor plugin". Read "NotePad++".

REFERENCES & RESOURCES

1. Microsoft MS-DOS User's Guide and Reference.
2. "Microsoft XP Professional Product Documentation - Command-line reference", [online].
3. "Microsoft XP Professional Product Documentation - Using batch file", [online].
4. "Command-line Reference for IT Pros - Technical Reference", available under Windows "Help".

Last modified: April, 2013