Java(tm) Tips and Tricks

Java(tm) Tips and Tricks

Submit your favorite tip or trick and win a Dippy-Bird T-shirt.
See details below:

See details below about how to enter.

Animation

- FlickerFree Animation
- Animated Applet on Colored Background

Audio

- AU Files for Java(tm) Audio Clips

Components

- Handling Double Clicking
- Creating a Read-Only TextArea
- TextField Selection Notification

File I/O

- Communicating with a CGI Program
- Checking for End of File
- Frames Support in Applets
- How to make an applet go to a URL when the mouse is clicked.

Graphics

- Java(tm) Dithering Explained
- Reading a Local JPEG File
- How to tell if an image is loaded
- Superimposing Text on Graphics

- Displaying a Sprite with Transparency

Java(tm) Language

- A Common Package Problem
- Differences between Paint, Repaint, and Update
- Passing Parameters by Reference
- String Equality
- Converting Strings to Char

Programming Style

- Variable Naming Conventions
- Debugging

Java(tm) IDE's

- Using Visual C++ for Java(tm) Development by Berkowitz and Carlson
- Using Diva(formally known as JavaSide) for Java(tm)
- CodeWright by Premia Corporation
- Other IDE's

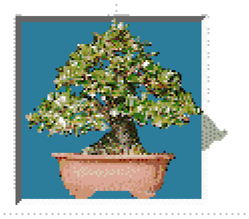
How to Enter
All you have to do is submit a tip or trick via e-mail before March 15th.

One name will be selected at random from the list of those who submitted a tip or trick. That winner will receive a Dippy-Bird T-shirt.

If you submit more than one tip or trick, your name will be put in the list twice.

Send entries to contest@dippybird.com   and please specify T-shirt size.

Animated Applet on Colored Background



Animated Applet on Colored Background

In order to animate an applet on a colored background, one way is to pass as a parameter the background color to the applet and expect the applet to paint the background color within the defined window.

This is reasonably trivial to do in the applet and very low overhead. Unfortunately, the downfall is to have to pass the background color to the applet as a parameter.

Back to Tips and Tricks

Communicating With a CGI Program

Communicating With a CGI Program

You can access a CGI script via it's URL by passing parameters after the '?' sign. That way, the CGI script gets the entire string after the '?' in the environment variable QUERY_STRING.

Here is an example:

```
  public boolean action(Event evt, Object arg) {
    if ("Search".equals(arg)) {
      URL doc = null;
      try {
        doc = new URL("http://www.somewhere.nl/htbin/Mycgiscript?Myparam="
                       + paramval);
      } catch (MalformedURLException e) {
        doc = null;
      }

      if (doc != null ) getAppletContext().showDocument(doc);
      return true;
    }
    return false;
  }
```

Back to Tips and Tricks

CodeWright 4.0

CodeWright by Premia Corporation

CodeWright's new programmer editor claims to have a Java(tm) error parser. It also, like Diva(tm) has HTML support built in.

One new feature I like which is not Java(tm) related, is its ability to let you insert graphical buttons in text files which perform special actions when you click them. This in essence gives you the ability to pup pop-up flow charts in your code.

Premia sales is at 1-800-547-9902

Back to Tips and Tricks

Debugging

Debugging

A good way to switch debug mode on and off is to include the following in your applet's tag definition.

```
<;param name=debug value="true">;
```

And then in your applet, check for the existence of the new switch like this:

```
Boolean debugMode;
String arg = getParameter("debug");
if (arg == null) {
    debugMode = false;
{ else {
    debugMode = true;
}
if (debugNode)
    showStatus("Program is Blah Blah Blah...");
```

Back to Tips and Tricks

Java Dithering Explained

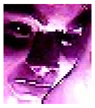
Java Dithering Explained

Below describes why Java may not be able to display all of the colors in an animation, resulting in an image that is dithered.

Java dithers all images on 8-bit displays to a standard color cube. Programs which deal with one image at a time can often customize the colormap for that one image, but Java has to contend with an arbitrary number of images with potentially disparate color needs so it chooses a standard set of colors which span the RGB spectrum and dithers to that standard set rather than a customized set for speed, robustness, simplicity of the APIs, and polite sharing of the colormap with other applictions. They may extend the APIs and the implementation in the future to offer higher quality color choosing algorithms, but the low price of the latest offering of 24-bit graphics cards in todays PC market may soon make such an effort pointless...

Back to Tips and Tricks

Using Diva(tm) as an IDE



Using Diva(tm) as an IDE

I personally recommend the Diva(tm) IDE.

Advantages are it takes a small footprint, it is very user friendly, and it has powerful features such as file find (grep).

Disadvantages are that it is currently in alpha stage, and there are a few bugs which you would expect with alpha software.

Currently (2-13-96), you can get Diva(tm) for evaluation purposes at http://www.inch.com/~friskel>

Back to Tips and Tricks

Handling Double Clicking

Handling Double Clicking

A very bright person on comp.lang.java explains it best.

There's no way for the toolkit (or the programmer!) to somehow intuit that the user of an app is planning on doing a double-click. It _must_ reflect every click to you -- or would you rather that every single click be delayed by 200 ms on the off-chance the user might be thinking about double clicking?

That's why every user-interface std out there mandates that double-clicking cause an extension of whatever single-click does. Hence the "1-click - insert here, 2-click - select word, 3-clicks - select line" behavior; each is an extension of the last.

Note that if single-click is meaningless, then you can get what you want simply by checking clickCount() and ignoring the event unless it's 2.

Another option is to "undo" whatever your single-click processing is if/when you notice the double click. Chances are good that your users will grumble abpout the non-intuitive interface, though. In GUI design, the principle of Least User Astonishment always applies.

Back to Tips and Tricks

AU Files for Java Audio Clips

AU Files for Java Audio Clips

Cool Edit version 1.50 is good for converting WAV to AU under Win95. Another utility is sox, however,cool edit is a full featured sound editor.

Back to Tips and Tricks

Checking for End Of File

Checking for End Of File

In Java, there is no keyword EOF or eof for checking for End Of File.

Instead, you should see how many bytes are availiable by using one of the stream classes.

Back to Tips and Tricks

Flicker Free Animation

Flicker Free Animation

Below is an excellent description by **kkessler** of how it is done.

---

1. Override Update()

2. Use Double Buffering

3. Override imageUpdate to notify your applet when the GIF has been loaded. This way the animation won't start before you image is downloaded.

4. Don't load each frame as a seperate GIF. Instead, create one GIF with all the frames in it.

You can find top notch Applets and source code on the following home page which shows how this is done:

http://www.access.digex.net/~kkessler

Back to Tips and Tricks

Frames Support in Applets

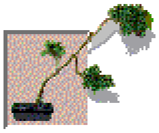
Frames Support in Applets

A very bright fellow named Eric Knudson has modified the ImageMap.class from Sun to additionaly take a TARGET name for frame URL's. It also will now take parms for single click vs. double click URL locations.

You should be able to find the ImageMap source on Gamelan (use search). .

Back to Tips and Tricks

How to make an applet go to a URL when the mouse is clicked



How to make an applet go to a URL when the mouse is clicked

Below describes How to make an applet go to a URL when the mouse is clicked.

Use showDocument (getAppletContext (), ...)

Back to Tips and Tricks

Reading a Local JPEG File

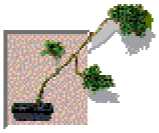
Reading a Local JPEG File

The trick is to use the Toolkit class which provides a method to read an image from a local file or a remote file via a URL. The format of the image can be GIF or JPG. Here is an example,

```
import java.awt.*;

public class testa extends Frame {
  Image image;

  public testa() {
    image =  Toolkit.getDefaultToolkit().getImage("c:\\image.jpg");
  }

  public void paint(Graphics g) {
    g.drawImage(image,1,1,this);
  }

  public static void main(String args[]) {
    testa t = new testa();
    t.pack();
    t.show();
  }

}
```

Back to Tips and Tricks

How To Tell if an Image is Loaded

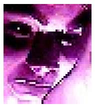


How To Tell if an Image is Loaded

This is easy. Simply use the java.awt.MediaTracker class.

Back to Tips and Tricks

Other IDE's



Other IDE's

Symmantec and Borland, both have add ons for their C++ IDE's.

However, currently, you must buy the C++ IDE in order to get the Java add on.

Borland also has a pre-release debugger that can currently (2-13-96) be downloaded from http:www.borland.com/Product/java/debugger/bugform.html

Programmers File Editor(PFE), while not an IDE, is an excellent freeware programmers editor for Windows 95 or Windows NT. I like to use this editor as a replacement for notepad, because it pops up really quick, but is loaded with functionality.

It can be downloaded at http://www.lancs.ac.uk/people/cpaap/pfe/pfefiles.htm .

Back to Tips and Tricks

Common Package Problem

Common Package Problem

The following is a common scenario of a problem that can happen with packages.

You have an applet "Bar" that worked fine until you add a "package foo;" statement at the top of the file. Now when you load the applet with Netscape you get the message **foo.Bar != Bar** on the status bar and the applet is not started. It also fails with the applet viewer.

Now here is the simple solution:

Put the file Bar.java in a subdirectory called "foo"

Recompile, as in" javac foo/Bar.java

Now "foo/Bar.class" should be able to load fine.

Back to Tips and Tricks

Difference Between Paint, Repaint and Update

Difference Between Paint, Repaint and Update

- paint - is a message/procedure that you write that willdraw whatever you want on the screen. An applet calls this function when the applet begins, and whenit is exposed (another window is moved off the applet). Don't call this message/procedure in your code.

- repaint - is the message/procedure that you call when you want the applet to redraw itself.

- update - is a message procedure that you may want to write. It is called after you call a repaint message. If you don't overload it, the default procedure is to clear the background to gray, and then call the paint procedure. For a smooth animation, you will want to overload this message.

Back to Tips and Tricks

Passing Parameters by Reference

Passing Parameters by Reference

You can update "object" parameters, but not parameters of a simple type.

Hence, to create an output parameter of a simple type, you must wrap it in an object, and pass the object. If you do this a lot, you may want to create a few "wrapper" classes, one for ints, one for floats, etc., each of which would have one public field of the "wrapped" type. The wrapper classes in the Java standard library aren't very useful for this, because you can't easily update their contents.

Back to Tips and Tricks

String Equality

String Equality

The following is a common scenario of a problem that can happen with string comparisons.

You have an applet in which you test the string by using

```
        stringA = xxx.readLine();
        if (stringA == "StringB")
```

Even though the two strings are equivalent, the comparison does not work.

Now here is the reason why:

The == operator is comparing what's in the variable stringA (which is an object reference or "pointer value") with the object "StringB".

It's NOT comparing the content of two strings to see if they are equal, it's asking, are they the same object?

Just because two strings have the same content doesn't mean they are the same object.

```
You should write:
    if (stringA.equals("StringB")) ..
```

The equals method for String compares content. For other classes (eg Rectangle) it does other things.

Back to Tips and Tricks

Converting Strings to Char

Converting Strings to Char

In order to parse a character out of a string, use the **toCharArray** method described in the java.lang.string package.

Back to Tips and Tricks

Superimposing Text on Graphics

Superimposing Text on Graphics

Below is code describing how to superimpose text on a graphic, using a selected font, at a specific location.

```
Image im;
String text = "some text given by user";

public void init() {
        im = getImage(new URL("http://image.url"));
}

public void paint(Graphics g) {
        g.drawImage(im, 0, 0, this);
        g.drawString(text, Xcoord, Ycoord);
}
```

Back to Tips and Tricks

Creating a Read-Only TextArea

Creating a Read-Only TextArea

Because, TextArea is a subclass of TextComponent you can use TextComponent.setEditable() to set the read-only properties of a TextArea control.

Back to Tips and Tricks

TextField Selection Notification

TextField Selection Notification

Because the native toolbox object eats it and AWT doesn't tell you, getting the TextField selection event is complicated. Extending won't help - **you can't get access** to something the existing AWT Peer doesn't give you.

Build your widget starting from Canvas - it's the only way to get control over mouse and key events.

Back to Tips and Tricks

Displaying a Sprite With Transparency in Java

Displaying a Sprite With Transparency in Java

As described by Bill Tschumy

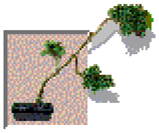
To display a sprite with transparency, just save your GIF designating one of the colors as transparent (you must use a pixel editor that supports that feature). Java reads it correctly and does indeed display the designated color transparently. Perfect for sprites.

Here is a program which saves transparent bitmaps.

LVIEW pro, shareware available from http://world.std.com/~mmedia The free version doesn't have the ability to save transparent GIF's, the pro version (around $30) does.

Back to Tips and Tricks

Variable Naming Conventions



Variable Naming Conventions

Although, this is not a programming standard or anything, it is common practice to name a component or applet's graphic variable to be "***g***" as the following example illustrates:

```
public void paint(Graphics g) {
    Rectangle r = bounds();

    g.drawImage(image, 0, 0, r.width, r.height, this);
}
```

Another common practice is to call the returned variable from getParameter()   "***arg***" if it is used as a temporary variable. Below is an example.

```
public void init() {
    String arg = getParameter("Color");
    if (arg !=null) {
        appColor = colorFromString(arg,Color.red);
    }
    arg = getParameter("image");
    if (arg !=null) {
        image = getImage(getDocumentBase(),arg);
    }
}
```

Back to Tips and Tricks

Using MSVC for Java

Using Microsoft Visual C++ for Java development

Introduction

This page has been put together from various contributions to the comp.lang.java newsgroup on the use of Microsoft Visual C++ as a Java IDE.

It turns out that the MSDEV/VC++ 4.0 tools (which We'll refer to as VC++ for short) can support Java development reasonably well. This note summarizes how customize VC++ for use with Java. We are sure that other improvements are possible; if you find any (or spot any errors in this page), please email them to andycarlson@attmail.com and this page will get updated.

These suggestions have been tried by the authors using Visual C++ 4.0. Some have been proven with JDK Beta 1 and others with JDK Beta 2.

Date: 18-Jan-96

Authors, and Acknowledgements:

This document was initially written by Jeff Berkowitz (jjb@netcom.com). It has been converted to HTML and added to by Andy Carlson of AT&;T (andycarlson@attmail.com) who is also the current maintainer of the document. First person singular pronouns, generally are Jeff talking.

Thanks also to all contributors to comp.lang.java on this subject.

Contents

Copyright

Disclaimer

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

You make these changes to your system at your own risk. Remember that you can really screw up your system with the Registry Editor!

Syntax coloring (or ";colouring"; for you native speakers.)

The editor can do a nice job of syntax coloring for Java source code. There are two steps to setting this up. First, the editor only applies global syntax coloring to ";C or C++ like"; files. So you need to teach it that Java files are ";C++ like."; Second, you need to teach it about the non-C++ keywords in the Java language.

Make sure VC++ is shut down, fire up your Registry Editor and dive down to:-

```
HKEY_CURRENT_USER

  Software

    Microsoft

      Developer

        Text Editor

          Tabs

            Language Settings

              C/C++
```

Now double click ";File Extensions."; You'll be rewarded with a dialog box that will allow you to edit the semicolon-separated list of extensions. Carefully add ";java"; to the end of the list. That's it. VC++ seems to support the four- character file extension just fine...at least We've seen no problems so far.

Note that this registry entry is not present until after you have edited and saved at least one C++ file after installing VC++ 4.0.

I believe the following is a list of the non-C++ keywords in Java. Save the list as a text-only file called USERTYPE.DAT in the same directory as your MSDEV.EXE (e.g. C:\MSDEV\BIN.) That's all there is to it. The next time you start Developer's Studio, these strings will be given the ";user defined type"; color, which happens to be the same as the keyword color (blue) by default. For more information, look at ";Syntax coloring user- defined types"; in the help. (And of course, you can add some words like true, false, and String that aren't strictly speaking ";keywords"; but are like them.)

```
----- Cut here and store as flat text in USERTYPE.DAT --------

abstract

boolean

byte

cast

extends

final

finally

future

generic

implements

import

inner

instanceof

native

null

outer

package

rest

super
```

```
synchronized

throws

transient

var

---------------------- End of USERTYPE.DAT -----------------------------
```

Some issues and limitations: These strings will be colored as keywords in *all* C++-like source files, now, not just .java files. USERTYPE.DAT is read only once when Developer Studio initializes, so you have to exit to change it. Alternatively, to disable syntax coloring of the Java extension keywords you can select ";Tools:Options...";, then the Format tab in the dialog box, scroll down to ";User Defined Keywords"; and set the color to black or whatever your normal text color is. Alternatively, select a colour (e.g. purple) which is different to both C++ keywords and normal text which may prevent confusion (but then again, it might not!).

Finally, there is only one class of user-defined symbols. Since we used it for keywords, we cannot e.g. color all the core and language names dark grey or anything cool like that.

We view these limitations as minor. And is very pleasing to see all those ";import"; and ";static final"; statements receiving MS VC++ syntax coloration... take that, you C++ thugs. ;-)

Compiling

The Java compiler can be added to the Tools menu. The Tools Configure... dialog has a flexible scheme for passing arguments to user-configured tool programs. More about the Tools menu below. Alternatively, you can also use the custom build feature to get VC++ to compile Java code which is in a VC++ project. See the section on Using Projects

We recommend as a minimum that you wrapped a little .bat file around the javac compiler so the message ";compilation complete"; would appear in the output window. Otherwise, there is no way to tell when the compiler is done after all those sucessful compilations :-) (by default, javac makes no output at all.).

If you want to go further, the F4 key can be made to work to allow you to step through build errors (I know: you never have any) and jump to the offending source line. To do this requires a perl script which filters the output of javac to look more like the Microsoft C++ compiler output. You will, of course, need a copy of perl. This can be found for Windows NT at http://info.hip.com/ntperl/

```
# javac.pl: perl script to run java compiler from MSVC 4.0

# Andy Carlson 1996

#

# Set up MSVC as follows (adjusting directories to suit):-

#
```

```
# Command: c:\winnt35\system32\cmd.exe
# Arguments: /c c:\win32app\perl5\bin\perl.exe c:\scripts\javac.pl $(FileName)$(FileExt)
# Directory: $(FileDir)
# Redirect to output window: Yes
# print all environment variables
#foreach $envvar (sort keys(%ENV))
#{
# print ";$envvar=$ENV{$envvar}\n";;
#}
# MSVC 4.0 loses all environment variables when running a tool,
# so CLASSPATH must be set up here
#
$ENV{";CLASSPATH";} = ";c:\\java\\classes";;
print ";CLASSPATH=";,$ENV{";CLASSPATH";},";\n";;
$wd = `cd`;
chop($wd);
print ";Compiling $ARGV[0]...\n";;
open (HIN,";c:\\internet\\java-beta\\bin\\javac $ARGV[0] |";) || die ";Can't start compiler";;
$ok = 1;
while (<;HIN>;)
{
# Turn this:-
# ClassA.java:3: Interface t813.IfA of class t813.ClassA not found.
# into something like this:-
# C:\java\browsetest.cpp(1) : error C2501: 'import' : missing decl-specifiers
if (/^((\D:)?[^:]+):(\d+):(.*)$/)
{
```

```
print ";$wd\\$1($3) : error J0000: $4\n";;
}
else
{
print;
}
$ok = 0 if (/^\d+ error/);
}
close(HIN);
print ";Compilation Successful.\n"; if ($ok);
```

See the note in the problems section for more info about tool spawning.

The obvious question has already been asked: 'Why perl and not java'. The answer is simple: the script was written at a time when its author was more familiar with perl than java. You are welcome to re-do the script in java if you feel the urge, but be warned that the script uses perl features which do not have immediately obvious equivalents in java.

Executing

The Java interpreter and applet viewer can also be added to the Tools menu in a similar fashion to the compiler. You could add Navigator to the Tools menu in order to test your applets from within a browser. More about the Tools menu below.

Using Projects

Visual C++ allows pretty much any text file to be included in one of its projects. To do this, create yourself a new Project Workspace with a type of Win32 Console App. Then you can add your Java source code to the project in the same manner as for C++. This gives you the ability to keep track of related sets of source code and remember window configurations. To take advantage of the project build facilities read on...

As an alternative (or in addition to) Compiling using the tools menu, you can get VC++ to build Java Source using the project file in a similar fashion to C++ source. To do this, you need to set up a Custom Build step for your Java Source Files. Once your source files are added, go into the Build/Settings option and select the Custom Build Tab. Highlight all the java sources and enter the following info:-

Description: 'Compiling Java Source...'
Build Command: cmd /c c:\win32app\perl5\bin\perl c:\scripts\javac.pl $(InputName).java
Output File: $(InputDir)\$(InputName).class

See the note in the problems section for more info about tool spawning.

Using this setting, as well as getting the F4 key working (using the perl script), you can build your whole project with a single keystroke, have dependencies managed, and VC++ will even count your errors at the end of the build and play the correct sound!

Source Code Browsing (not to be confused with web browsing)

The Visual C++ 4.0 Source Browser will not quite work with Java source code, even when you tell it that a .java file is a C++ source. As far as I (Andy) can tell, it gets confused (not surprisingly) by the following:-

- The public keyword preceding the class keyword or a class member definition
- The java inheritance syntax confuses the browser into using the base class name instead of the class being defined.

Attempts to fool the browser such as interposing line breaks and/or closed 'C' comments between the offending keywords have so far failed. The browser will, I think work with private classes and their members which are not preceded by an access specifier.

[Note: my (Jeff) employer has a site license for the ";cscope"; program, an old but very useful source code browser from the ex-AT&;T Unix Toolkit. I ported ";cscope"; from Unix to Windows95 and made a few hacks so it would work with VC++ 4.0. I cannot redistribute it, though; don't even ask.

So I took out this section of the instructions from this net post. If somebody can propose a source code browser, I (Jeff) can help explain how it could be integrated. It would be best if it could be run as a console child program, producing line-oriented output.]

Tools Menu Summary

Menu text: &;Applet Viewer
Command: C:\jdk_b1\java\bin\appletviewer.exe
Arguments: $(FileName).html
Directory: $(FileDir)

Menu text: &;Compile Java Source
Command: C:\jdk_b1\java\bin\javac.exe
Arguments: $(FileName)$(FileExt)
Directory: $(FileDir)
Redirect to output window: Yes

Menu text: &;Java Compiler (with output filtering)
Command: c:\winnt35\system32\cmd.exe
Arguments: /c c:\win32app\perl5\bin\perl.exe c:\scripts\javac.pl $(FileName)$(FileExt)
Directory: $(FileDir)
Redirect to output window: Yes

Menu text: Java &;Interpreter
Command: C:\jdk_b1\java\bin\java.exe
Arguments: $(FileName)
Directory: $(FileDir)

Tabs

The Java source code uses a common Unix style of formatting: eight space tabs with a four space indent. The source code I looked at was not consistent, sometimes using tabs and sometimes using lots of spaces. The VC++ default seems to be four space tabs. You can change this in ";Tools:Options..."; on the ";Tabs"; tab. Be sure you change it for ";language C/C++"; since VC++ now thinks Java is a kind of C/C++ language.

Find in Files...

I discovered that you can customize the default set of file extensions to be searched when the Find in Files... dialog is first brought up.

It is stored in:-

```
HKEY_CURRENT_USER

  Software

    Microsoft

      Developer

        Search
```

and is called FIFFilter. Double click this semicolon-separated list and carefully add ";.java"; to it (or simply replace the string with ";.java"; if you are really through with C++ ;-)

Things not figured out yet

- The File Open... dialog comes up with the filter set to ";common files."; The Common Files string obviously doesn't include ";.java";. We haven't figured out to change this, and it is a significant hassle.
- Portable source code browser. Suggestions?
- It seems that MSVC 4.0 under NT 3.5.1 (and the new shell) spawns tools differently from MSVC 2.x as it now loses all environment variables and for some reason causes a problem starting the compiler from a perl script unless run from a cmd.exe shell. Windows 95 does not require the addition of cmd.exe (or rather command.com). Anyone know of a more elegant solution??

Alternatives

If you don't like VC++ as a Java IDE, don't have VC++ or just want a change, then you could try 'Javaside', a prototype IDE for Java and HTML by Jim Friskel. This does most of the things described for VC++ above, along with providing a simple source browser.

Javaside can be found at http://www.inch.com/~friskel/

URL Not Available

http://info.hip.com/ntperl/

URL Not Available

http://www.access.digex.net/~kkessler

URL Not Available

http://www.inch.com/~friskel

URL Not Available

http://www.inch.com/~friskel/

URL Not Available

http://www.lancs.ac.uk/people/cpaap/pfe/pfefiles.htm

URL Not Available

http:/www.borland.com/Product/java/debugger/bugform.html

URL Not Available

mailto:contest@dippybird.com

URL Not Available

news:/comp.lang.java