

# *Notes from Well House Consultants*

*These notes are written by Well House Consultants and distributed under their Open Training Notes License. If a copy of this license is not supplied at the end of these notes, please visit*

*<http://www.wellho.net/net/whcotnl.html>  
for details.*

## 1.1 Well House Consultants

Well House Consultants provides niche training, primarily but not exclusively in Open Source programming languages. We offer public courses at our training centre and private courses at your offices. We also make some of our training notes available under our "Open Training Notes" license, such as we're doing in this document here.

## 1.2 Open Training Notes License

With an "Open Training Notes License", for which we make no charge, you're allowed to print, use and distribute these notes provided that you retain the complete and unaltered license agreement with them, including our copyright statement. This means that you can learn from the notes, and have others learn from them too.

You are NOT allowed to charge (directly or indirectly) for the copying or distribution of these notes, nor are you allowed to charge for presentations making any use of them.

## 1.3 Courses presented by the author

If you would like us to attend a course (Java, Perl, Python, PHP, Tcl/Tk, MySQL or Linux) presented by the author of these notes, please see our public course schedule at

<http://www.wellho.net/course/index.html>

If you have a group of 4 or more trainees who require the same course at the same time, it will cost you less to have us run a private course for you. Please visit our onsite training page at

<http://www.wellho.net/course/otc.html>

which will give you details and costing information

## 1.4 Contact Details

Well House Consultants may be found online at

<http://www.wellho.net>

[graham@wellho.net](mailto:graham@wellho.net)

technical contact

[lisa@wellho.net](mailto:lisa@wellho.net)

administration contact

Our full postal address is

404 The Spa

Melksham

Wiltshire

UK SN12 6QL

Phone +44 (0) 1225 708225

Fax +44 (0) 1225 707126

# *Internationalization*

If you're going to be writing your application for worldwide use, you need to consider how you're going to provide your messages in different languages, how you're going to handle different character sets, and how you're going to handle different representations of dates, currencies and numbers. Java provides the tools, and we describe them in this module.

<i>Local Representations</i> . . . . .	4
<i>Unicode</i> s . . . . .	4
<i>Resource bundles</i> . . . . .	6

## 2.1 Local Representations

The way that dates, currencies, large numbers, etc are represented in the country of interest.

How many shares have you?  
What's the date? What's their value?

```
import java.text.*;
import java.util.*;

public class pintail
{
    public static void main (String[] args)
    {

        if (args.length == 2)
            Locale.setDefault(new Locale(args[0],args[1]));

        int shares = 6500;
        Date today = new Date();
        double value = 17.65;

        NumberFormat number =
        NumberFormat.getInstance();
        NumberFormat price =
        NumberFormat.getCurrencyInstance();
        DateFormat day =
        DateFormat.getDateInstance(DateFormat.SHORT);

        System.out.println
        ("Number of shares "+number.format(shares));
        System.out.println
        ("Value each today "+price.format(value));
        System.out.println
        ("Date "+day.format(today));

    }
}
```

Figure 1 Running public class pintail

```
seal% java pintail
Number of shares 6,500
Value each today $17.64
Date 4/18/98
seal% java pintail fr FR
Number of shares €00
Value each today 17,64 F
Date 18/04/98
seal% java pintail en GB
Number of shares 6,500
Value each today £17.64
Date 18/04/98
seal% java pintail fr CA
Number of shares €00
Value each today 17,64 $
Date 98-04-18
seal%
```

## 2.2 Unicodes

Handling of special characters (there is a huge set!).

Unicodes expressed as `\u22a0` and similar (hex)

Trivial mapping - `\u00xx` is `xx` ASCII

Remember that many characters are missing from many fonts.

Java uses Unicodes internally; they are a 16-bit character set, which means you have a large range.

Streams are unicode-based, but byte files are byte-based.

We are using the character-by-character drawing method `drawsChars` as a convenience. We could equally use `drawString`.

## Note definition of our own colour.

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.lang.*;

public class buzzard extends Applet {
    int charval = 0x2200;
    public static Font use;

    public void init() {

        use = new Font("Serif",Font.PLAIN,18);

        this.addMouseListener(
            new MouseAdapter() {
                public void mousePressed(MouseEvent e) {
                    Graphics g = getGraphics();

                    int window_width = getSize().width-1;
                    int window_height = getSize().height-1;
                    int mods = e.getModifiers();
                    if ((mods & InputEvent.BUTTON3_MASK) != 0)
                        charval -= 0x200;
                    if ((mods & InputEvent.BUTTON2_MASK) != 0)
                        charval = 0x00;
                }
            }
        );
    }
}

```

## Note sensing of mouse buttons

```

String base = Integer.toHexString(charval);
char [] display = new char[1];
g.setColor(Color.black);
g.fillRect(0,0,window_width,window_height);
Color mustard = new Color(180,180,50);
g.setColor(mustard);
g.setFont(use);
for (int k=0;k<16;k++) {
    for (int l=0;l<16;l++) {
        display[0] = (char) ++charval;
        g.drawChars(display,0,1,10+window_width/18*k,
            30+window_height/18*l);
    }
}
g.drawString(base,10,window_height-10);
}
});
}
}

```

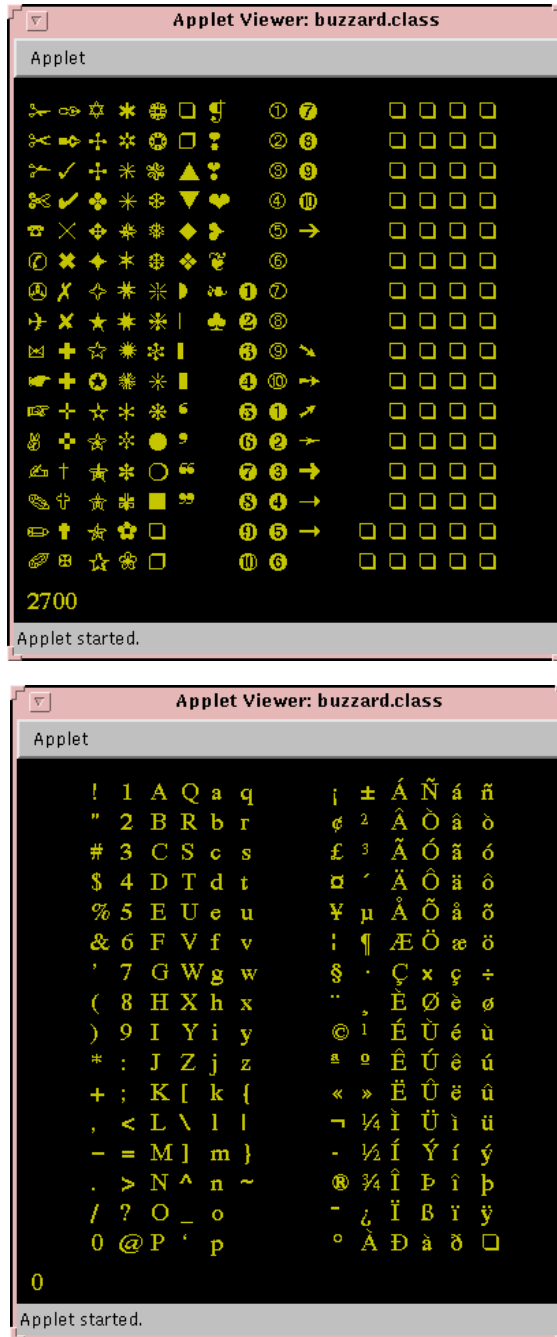


Figure 2 Two views of running public class buzzard in our AppletViewer

### 2.3 Resource bundles

Translation of all messages into the local language (and if necessary, sub-language such as "english" v "american").

We need to have it come out in the "right language".

Our *avocet* example prints out the first message the wrong way, then does the rest of its work through a resource bundle so that the resource for the correct country can be used.

Like in Localisation, we can specify a language and a substrain.

All the messages will be held in properties files, which should be provided for each supported language. If there is no message provided, the system will fall back to the default.

```

import java.util.*;

public class avocet
{
    public static void main (String[] args)
    {

System.out.println("How NOT to do it for use in
    France!  ...");

        if (args.length == 2)
Locale.setDefault(new Locale(args[0],args[1]));
        ResourceBundle rb = null;
        try {
            rb = ResourceBundle.getBundle("Avocet");
        } catch (MissingResourceException e) {
            System.out.println(
                "err - don't know your language");
        }
        try {
            String greeting = rb.getString("greeting");
            System.out.println(greeting);

            System.out.println(rb.getString("sentence"));
        } catch (Exception e) {
            System.out.println(
                "err - don't know that in your language");
        }
    }
}

```

Figure 3 Running public class avocet

```

seal% java avocet
How NOT to do it for use in France!
Hi there!
It's raining again
seal% java avocet en GB
How NOT to do it for use in France!
Good Morning
It's raining again
seal% java avocet fr FR
How NOT to do it for use in France!
Bonjour
Encore il pleut!
seal%

```

Capitalizing Avocet is necessary.

```

seal% more Avo*ies
::::::::::::
Avocet.properties
::::::::::::
# The default is US English
greeting=Hi there!
sentence=It's raining again
::::::::::::
Avocet_en_GB.properties

```

```
.....  
# Overrides for UK English  
greeting=Good Morning  
.....  
Avocet_fr_FR.properties  
.....  
# Override (everything!) for French  
greeting=Bonjour  
sentence=Encore il pleut!  
seal%
```

Note file naming.

Note ability to provide a partial or whole bundle.

# *License*

*These notes are distributed under the **Well House Consultants Open Training Notes License**. Basically, if you distribute it and use it for free, we'll let you have it for free. If you charge for its distribution of use, we'll charge.*

### 3.1 Open Training Notes License

Training notes distributed under the **Well House Consultants Open Training Notes License** (WHCOTNL) may be reproduced for any purpose PROVIDE THAT:

- This License statement is retained, unaltered (save for additions to the change log) and complete.
- No charge is made for the distribution, nor for the use or application thereof. This means that you can use them to run training sessions or as support material for those sessions, but you cannot then make a charge for those training sessions.
- Alterations to the content of the document are clearly marked as being such, and a log of amendments is added below this notice.
- These notes are provided "as is" with no warranty of fitness for purpose. Whilst every attempt has been made to ensure their accuracy, no liability can be accepted for any errors of the consequences thereof.

Copyright is retained by Well House Consultants Ltd, of 404, The Spa, Melksham, Wiltshire, UK, SN12 6QL - phone number +44 (1) 1225 708225. Email contact - Graham Ellis (graham@wellho.net).

Please send any amendments and corrections to these notes to the Copyright holder - under the spirit of the Open Distribution license, we will incorporate suitable changes into future releases for the use of the community.

If you are charged for this material, or for presentation of a course (Other than by Well House Consultants) using this material, please let us know. It is a violation of the license under which this notes are distributed for such a charge to be made, except by the Copyright Holder.

If you would like Well House Consultants to use this material to present a training course for your organisation, or if you wish to attend a public course is one is available, please contact us or see our web site - <http://www.wellho.net> - for further details.

Change log  
Original Version, Well House Consultants, 2004

Updated by: \_\_\_\_\_ on \_\_\_\_\_  
Updated by: \_\_\_\_\_ on \_\_\_\_\_  
Updated by: \_\_\_\_\_ on \_\_\_\_\_  
Updated by: \_\_\_\_\_ on \_\_\_\_\_  
Updated by: \_\_\_\_\_ on \_\_\_\_\_  
Updated by: \_\_\_\_\_ on \_\_\_\_\_

*License Ends.*