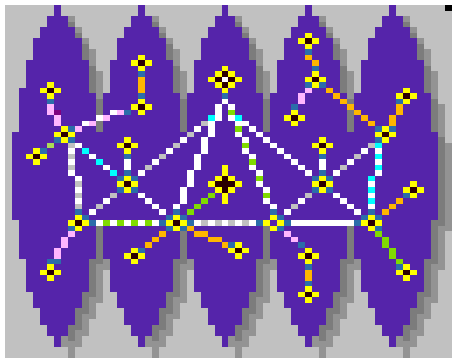
Doing GUI Programming with NetRexx on VM/ESA

Session M65

Chuck Morse
IBM Washington Systems Center
morsec@us.ibm.com



RETURN TO INDEX

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "AS IS" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environments.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Trademarks

The following are trademarks of International Business Machines Corporation. Those identified with an (*) are registered trademarks of International Business Machines

- Enterprise Systems Architecture/370
- Enterprise Systems Architecture/390
- ESA/370
- ESA/390
- IBM*
- System/370
- VM/ESA*
- OpenEdition

The following are trademarks of the corporations identified

- Windows 95 - is a registered trademark of Microsoft Corporation
- JAVA - is a trademark of Sun Microsystems
- "Write once, run anywhere" - is a trademark of Sun Microsystems

Agenda

- **An Introduction to Java and NetRexx**
- **Java support on VM/ESA**
- **The Remote Abstract Windowing Toolkit**
- **Abstract Windowing Toolkit Examples**
- **Comparison to other Graphical User Interfaces on VM/ESA**

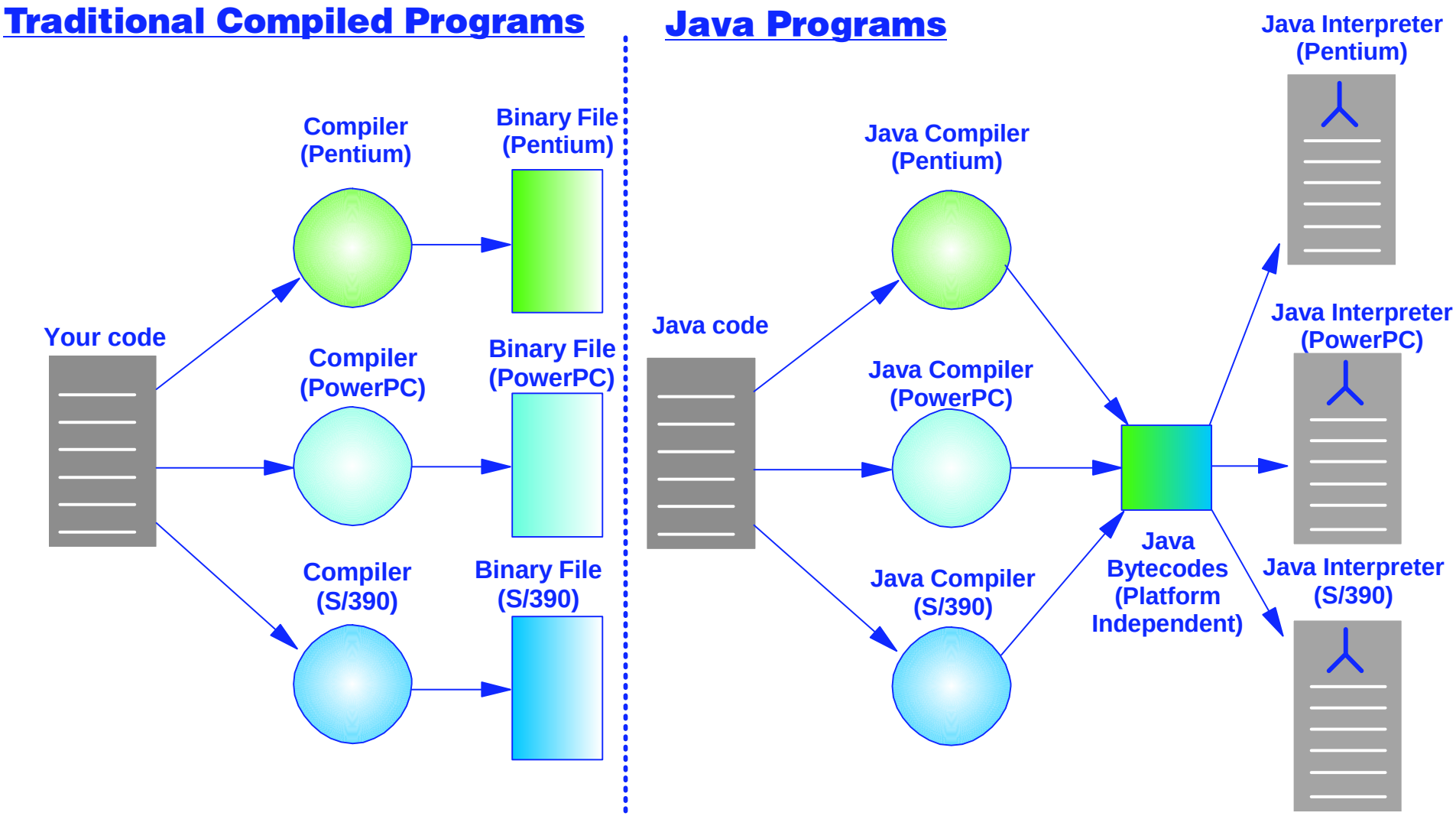
The Java Language

- **A simpler, safer dialect of C and C++ with things that are complex, dangerous, or error-prone eliminated**
  - No preprocessor, pointers, operator overloading, etc.
- **Object oriented with a rich set of classes**
  - Supports single inheritance
  - Networking, Threading, Windowing ...
- **Java compiler does extensive checking**
- **Write once, run anywhere philosophy**

The Java Virtual Machine

- **A software microprocessor with its own instruction set**
- **The JVM interprets bytecodes produced by the Java compiler**
  - Architecture independent
- **The JVM performs run time checking**
  - Type and bounds checking
  - File I/O errors, exceptions
- **Just-in-time (JIT) compilation may improve efficiency**

The Java Virtual Machine Concept

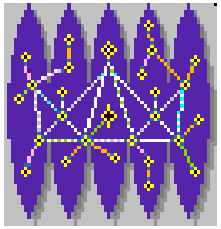


The Java Developer's Kit

- Packages (class libraries)
  - Common to every implementation
  - Java source included
    - java.lang, java.util, java.math, java.text [strings, numbers, date/time...]
    - java.io, java.net [file and network I/O]
    - java.awt [abstract window toolkit], java.applet [animation, audio]
    - java.security [public keys, cryptography]
    - java.sql [database], java.rmi [remote methods]
    - java.beans [library of pluggable components]
- Programs
  - javac — Compiles Java source into bytecodes
  - java — Invokes the JVM to run a compiled application
  - appletviewer — Previews a compiled applet
  - javadoc — Extracts interface documentation from source
  - javah — Generates C skeletons for native methods
  - javap — Disassembles Java class files
  - jdb — Runs the Java debugger
- Samples and demos to illustrate usage

The NetRexx Lineage

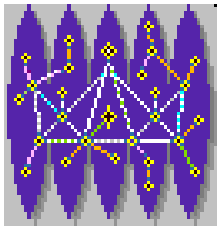
- Rexx is easy to use for scripting
  - Simple, readable syntax
  - Natural data typing
  - Decimal arithmetic
  - Safe for interpretation
- Java is robust for application development
  - Object oriented
  - Strong or static typing
  - Compilable
  - Portable



- NetRexx is a blend of Rexx and Java
  - Developed by IBM Fellow Mike Cowlishaw
  - Rexx syntax + Java object model
  - Rexx ease of use + Java portability
  - Automated type selection and declaration
  - High productivity and simplicity for users
  - A truly general-purpose language
  - "Designed for users, not compilers"
  - Available on web (see Appendix)

Interoperability with Java

- **NetRexx source translated to Java source code**
  - The translator is itself a Java program
  - Portable class files are assured
  - Built-in or external Java compiler may be used
  - Runs wherever there is a JVM
- **Any class written in Java can be used from NetRexx**
  - Type conversions are generally automatic
- **Any class written in NetRexx can be used from Java**



NetRexx Illustrative Examples

JAVA

```
HelloWorld.java
  public class HelloWorld {
    public static void main(String[ ] args {
    System.out.println("Hello World!");
    }
  }
```

NETREXX

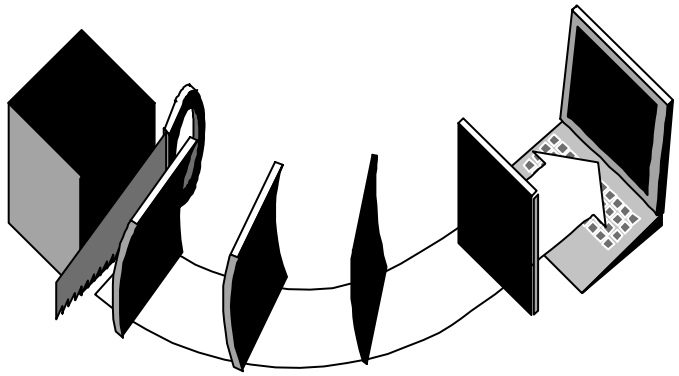
```
Toast.nrx
  /* This wishes you good health. */
  say "Cheers!"
```

Java & NetRexx on VM/ESA

- **Requires VM/ESA V2R3.0+**
  - Byte File System
  - OpenEdition Shell and Utilities
- **Java/NetRexx download instructions are at:**
  - http://www.ibm.com/s390/vm/java/
- **Available today via FTP**
  - NetRexx 1.160
  - The Java Compiler
  - The Java Debugger
  - The Java Virtual Machine
  - The Sun Microsystems' JDK 1.1.6 class libraries
    - Year 2000 ready
  - The Java Native Interface (JNI)
  - A Just-In-Time compiler (JIT)
  - Remote Abstract Windowing Toolkit (RAWT)
-

Remote Abstract Windowing Toolkit

- **What is it?**
  - Implementation of all AWT APIs In a client/server fashion transparent to Java application
    - Graphics services carried out on remote workstation
  - Allows the running of graphic Java applications on a powerful host that does not support graphics
  - No modifications required to graphics application
- **Requirements:**
  - TCP/IP on host and workstation
  - JDK 1.1.x on host and workstation
  - Workstation with graphics support



Installing the Remote AWT

- **RAWT host code is preinstalled with JDK 1.1.6 for VM/ESA**
  - Class files located in $JAVA_HOME/lib/RAWTApplHost.zip
  - See $JAVA_HOME/RAWT-readme.html for further info
- **Obtain workstation code from same location:**
  - Unzip or tar the workstation code following instructions in RAWT-readme.html

Installing the Remote AWT ....

- **Build host shell script for starting applications**

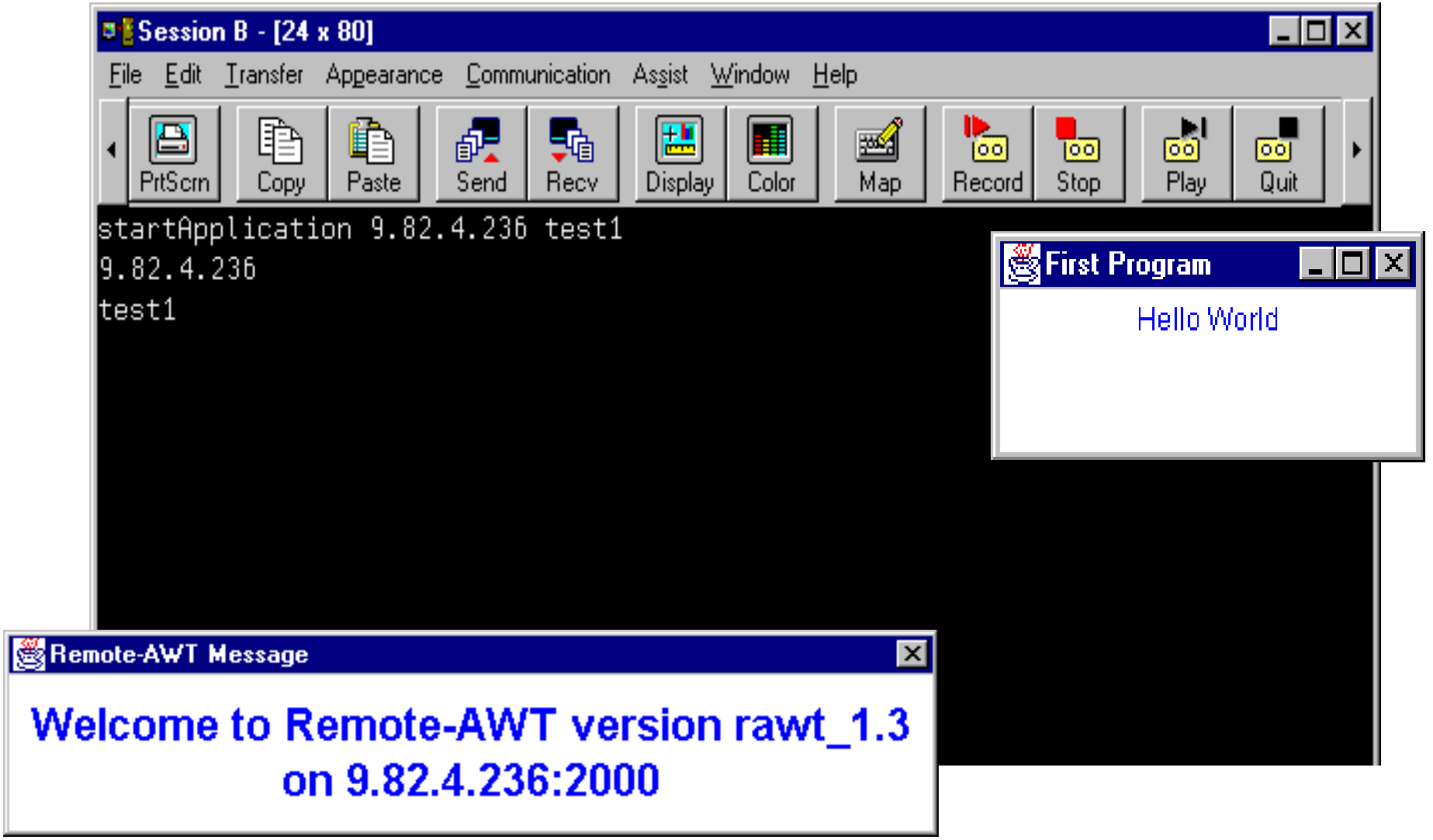
```
#REM  syntax:
#REM  java  <-DJdkVersion=xxxxx>
#REM        -Dawt.toolkit=com.ibm.rawt.client.CToolkit
#REM        -DRmtAwtServer=UserStationHostnameOrIPAddress
#REM        ApplicationName <arguments for ApplicationName>
echo $1
echo $2
export \
CLASSPATH=$JAVA_HOME/lib/RAWTApplHost.zip:$JAVA_HOME/classes:$CLASSPATH
$JAVA_HOME/bin/java -Dawt.toolkit=com.ibm.rawt.client.CToolkit  \
                    -DRmtAwtServer=$1 $2 $3 $4 $5 $6
```

- **chmod +x scriptname to make it executable**
- **On VM host make sure DNS configured in TCPIP DATA or HOST tables available that specify IP address for host name listed in TCPIP DATA**
  - Java networking package insists host be known by name

Using the Remote AWT

- **Start RAWT on the workstation**
  - Windows:
    - set RAWT_HOME=c:\RAWT
    - set PATH=%PATH%;%RAWT_HOME%\bin
    - runRAWT
- **Start Java application on VM/ESA host**
  - From within the shell type:
    - yourScriptName x.x.x.x application_name
    - Where: x.x.x.x is the IP address of the target workstation
    - Where: application_name is the Java class file to start (without specifying the .class extension)
- **Application window will appear on your workstation**
- **Ending application on workstation will end application on host (startApplication will complete, back to shell prompt)**

Remote AWT Example

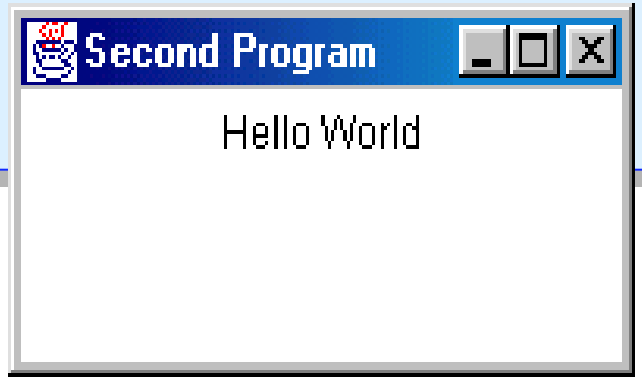


Hello World using AWT

```
 /*Test 1 - Hello World  */
class test1
  Properties inheritable
    window = Frame
  method main( args=String[] ) public static
    test1()  -- Creates an object of the class test1
  method test1() -- Create a frame window
    window = Frame('First Program')
    window.setSize(250,150) -- Set the window size
    window.addWindowListener( CloseWindowAdapter() )
-- Put the text 'Hello World in the center top of window
    window.setLayout(BorderLayout())
    banner = Label( "  Hello World  ",label.center)
    window.add("North",banner)
-- show the window
    window.setVisible(1)
class CloseWindowAdapter extends WindowAdapter
  method windowClosing( e=WindowEvent )
    exit 0
```

Centering the Window

```
 /* Test2 - Center the Window */
class test2
  Properties inheritable
    window = Frame
  method main( args=String[] ) public static
    test2()  -- Creates an object of the class test1 and shows the window
  method test2()  -- Create a frame window
    window = Frame('Second Program')
    window.setSize(250,150)  -- Set the size of the window
-- Set the window position to the middle of the screen
    d = window.getToolkit().getScreenSize()
    s = window.getSize()
    window.setLocation((d.width - s.width) % 2,(d.height - s.height)%2)
-- add the window event listener to the window for close window events
    window.addWindowListener( CloseWindowAdapter() )
    ....
    ....
    ....
```

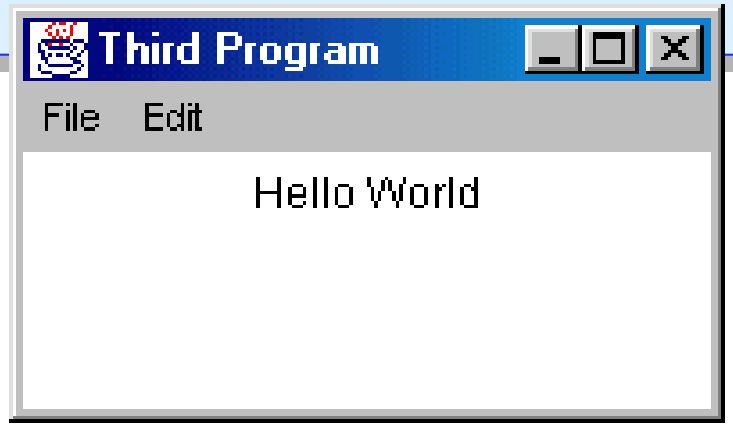


Adding a Menu Bar

```
/* Test3 - Adding a Menu Bar (page 1) */
class test3
  Properties inheritable
    window = Frame
  method main( args=String[] ) public static
    test3()  -- Creates an object of the class test1 and shows the window
  method test3()  -- Create a frame window
    window = Frame('Third Program')
    window.setSize(250,150)  -- Set the size of the window
-- Set the window position to the middle of the screen
    d = window.getToolkit().getScreenSize()
    s = window.getSize()
    window.setLocation((d.width - s.width) % 2,(d.height - s.height)%2)
-- add the window event listener to the window for close window events
    window.addWindowListener( CloseWindowAdapter() )
-- Put the text 'Hello World in the center top of window
    window.setLayout(BorderLayout())
    banner = Label( "  Hello World  ",label.center)
    window.add("North",banner)
    ...
```

Adding a Menu Bar (cont.)

```
/* Test3 - Adding a Menu Bar (page 2)*/
-- add a menu bar to the window
    mb=menuBar()
    mfile=Menu('File')
    medit=Menu('Edit')
    mb.add(mfile)
    mb.add(medit)
    window.setMenuBar(mb)
-- show the window
    window.setVisible(1)
class CloseWindowAdapter extends WindowAdapter
  method windowClosing( e=WindowEvent )
    exit 0
```

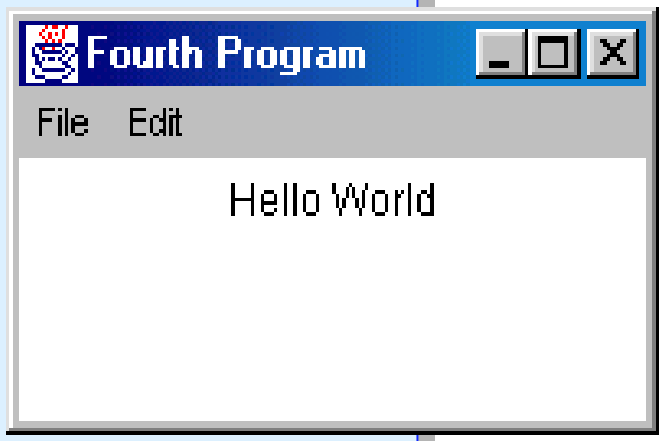


Adding Menu Items

```
/* Test4 - Adding Menu Items (Page 1) */
class test4
  Properties inheritable
    window = Frame
    mopen=MenuItem('Open')
    msave=MenuItem('Save')
    mcopy=MenuItem('Copy')
    mpaste=MenuItem('Paste')
    mexit=MenuItem('Exit')
  method main( args=String[] ) public static
    test4()  -- Creates an object of the class test1 and shows the window
  method test4()  -- Create a frame window
    window = Frame('Fourth Program')
    window.setSize(250,150)  -- Set the size of the window
-- Set the window position to the middle of the screen
    d = window.getToolkit().getScreenSize()
    s = window.getSize()
    window.setLocation((d.width - s.width) % 2,(d.height - s.height)%2)
```

Adding Menu Items (cont.)

```
/* Test4 - Adding Menu Items (page 2)*/
-- add the window event listener to the window for close
window events
    window.addWindowListener( CloseWindowAdapter() )
-- Put the text 'Hello World in the center top of window
    window.setLayout(BorderLayout())
    banner = Label( "  Hello World  ",label.center)
    window.add("North",banner)
-- add a menu bar to the window
    mb=menuBar()
    mfile=Menu('File')
    medit=Menu('Edit')
    mfile.add(mopen)
    mfile.add(msave)
    mfile.addSeparator()
    mfile.add(mexit)
    medit.add(mcopy)
    medit.add(mpaste)
    mb.add(mfile)
    mb.add(medit)
    window.setMenuBar(mb)
    ...
```

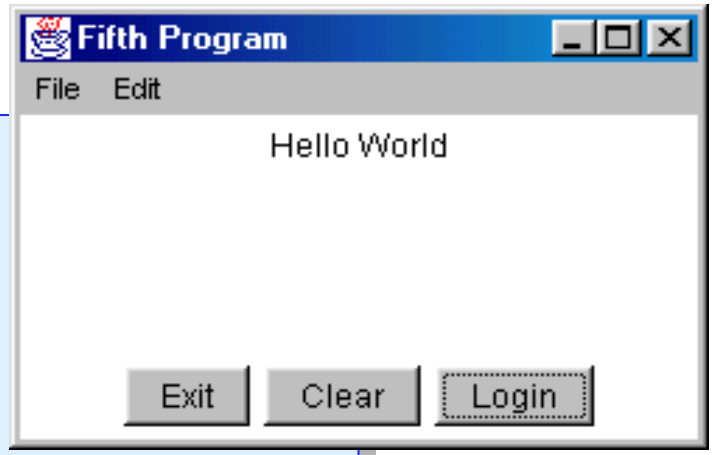


Adding Some Buttons

```
 /* Test5 - Adding some Buttons */
class test5
Properties inheritable
    window = Frame
    mopen=MenuItem('Open')
    msave=MenuItem('Save')
    mcopy=MenuItem('Copy')
    mpaste=MenuItem('Paste')
    mexit=MenuItem('Exit')
    bexit = Button("  Exit  ")
    bclear = Button("  Clear  ")
    blog = Button("  Login  ")
  method main( args=String[] ) public static
....
```

Adding Some Buttons (cont.)

```
....
    window.setMenuBar(mb)
    p1 = Panel()
    p1.add(bexit)
    p1.add(bclear)
    p1.add(blog)
    window.add("South", p1)
-- show the window
    window.setVisible(1)
class CloseWindowAdapter extends WindowAdapter
  method windowClosing( e=WindowEvent )
    exit 0
```

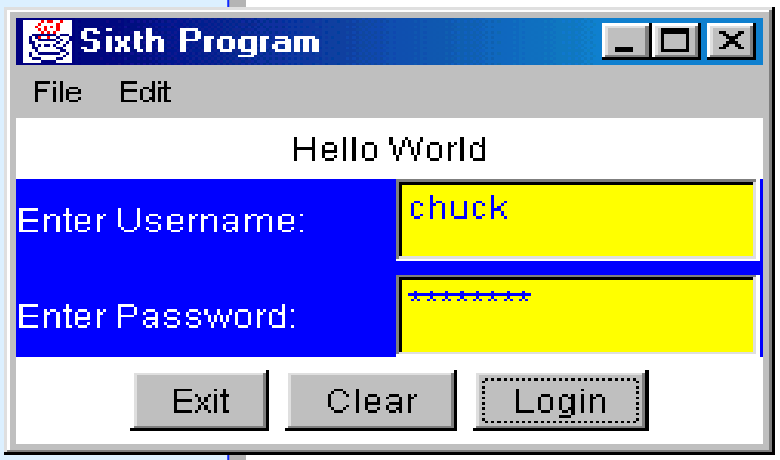


Adding Text Input Fields

```
/* Test6 - Adding Text Input Fields */
class test6
  Properties inheritable
    window = Frame
    mopen=MenuItem('Open')
    msave=MenuItem('Save')
    mcopy=MenuItem('Copy')
    mpaste=MenuItem('Paste')
    mexit=MenuItem('Exit')
    bexit = Button("  Exit  ")
    bclear = Button("  Clear  ")
    blog = Button("  Login  ")
    name = TextField(25)
    pw = TextField(25)
  method main( args=String[] ) public static
....
```

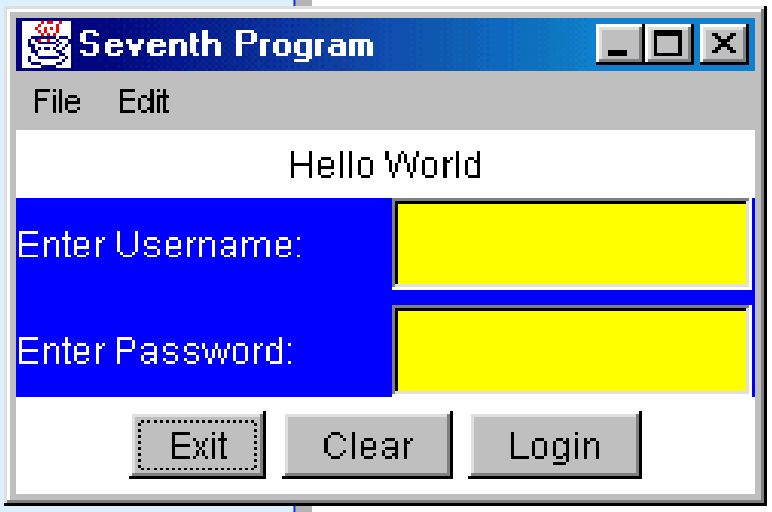
Adding Text Input Fields (cont.)

```
/* Test6 - Adding Text Input Fields (page 2)*/
....
    window.add("South", p1)
    p2 = Panel()
    p2.setLayout(GridLayout(2, 2, 5, 5))
    p2.setBackground(Color.blue)
    p2.SetForeground(Color.white)
    ulabel = Label("Enter Username:")
    plabel = Label("Enter Password:")
    name.setBackground(Color.yellow)
    name.SetForeground(Color.blue)
    pw.setBackground(Color.yellow)
    pw.SetForeground(Color.blue)
    pw.setEchoChar(char "*")
    p2.add(ulabel)
    p2.add(name)
    p2.add(plabel)
    p2.add(pw)
    window.add("Center", p2)
-- show the window
...
```



Making Things Happen

```
/* Test7 - Making things happen  */
class test7 implements ActionListener
....
    window.add("Center", p2)
-- Listen for Action Events
    mexit.addActionListener(this)
    bexit.addActionListener(this)
    bclear.addActionListener(this)
-- show the window
    window.setVisible(1)
method actionPerformed(e = ActionEvent)
    select
      when e.getsource = mexit then exit
      when e.getsource = bexit then exit
      when e.getsource = bclear then do
        name.setText('')
        pw.setText('')
      end
      otherwise nop
    end
class CloseWindowAdapter extends WindowAdapter
  method windowClosing( e=WindowEvent )
    exit 0
```



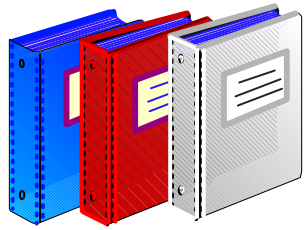
Graphical User Interfaces for Existing VM Applications

- **Screen Scraping Technology**
  - no change to application
  - normally must tailor screen scraper to the application
  - application dependent code must reside on each workstation
- **Web Technology**
  - No change to application
  - Easy to interface to existing applications using familiar CMS tools (Pipelines & Rexx)
  - Industry standard workstation agent (Web Browsers)
  - Easy to build new applications as well

Graphical User Interfaces for New VM Applications

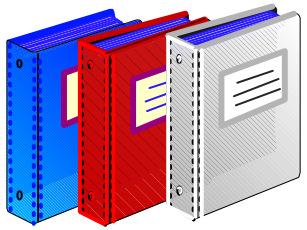
- **CMS GUI**
  - For new applications in C, C++, Rexx or Assembler
  - Common workstation agent on Windows, AIX or OS/2
    - No application dependent code on workstation
  - Communication via TCP/IP or APPC (Win 3.1 or OS/2 only)
  - High degree of difficulty
    - No GUI builder
- **Remote AWT with Java or NetRexx**
  - For new applications in Java or NetRexx
  - Allows porting of existing Java or NetRexx applications
  - Common workstation agent on any platform that supports JAVA
    - No application dependent code on workstation
  - Communication via TCP/IP
  - Several existing Java GUI builders
  - NetRexx makes coding GUI applications easy
  - High application startup costs
    - requires OpenEdition & BFS

Sources of Additional Information on the Web



- **IBM Centre for Java Technology Development**
  - http://ncc.hursley.ibm.com/javainfo/
    - Current information on Java-related activities
    - Downloadable code packages and links to other sites
- **Mike Cowlishaw's NetRexx Language page**
  - http://www2.hursley.ibm.com/netrexx/
    - Latest NetRexx documentation and status
    - Downloadable code packages and links to other sites
- **IBM's Redbook page: http://www.redbooks.ibm.com**
  - *Creating Java Applications Using NetRexx*, SG24-2216-00
  - *VM/ESA Network Computing with Java and NetRexx*, SG24-5148-00
- IBM's Java page: **http://www.ibm.com/java**
- Sun's Java page: **http://www.javasoft.com**
- The VM Java page: **http://www.vm.ibm.com/java**

Sources of Additional Information in Print



- **Mike Cowlishaw, *The NetRexx Language***
  - ISBN 0-13-806332-X, or IBM Puborder SR23-7771
- **Ken Arnold and James Gosling, *The Java Programming Language***
  - ISBN 0-201-63455-4
- **David Flanagan, *Java in a Nutshell***
  - ISBN 1-56592-183-6
- **Peter van der Linden, *Just Java***
  - ISBN 0-13-565839-X
- **J. Savit, S. Wilcox, B. Jayaraman, *Enterprise Java***
  - ISBN 0-07-057991-1