Frequently Used Java Commands

April 4, 2023

Native Java Documentation

Integer

```
String str = Integer.toString(iValue);
String str = Integer.toHexString(iValue);
int iValue = (Integer.valueOf(str));
```

Double

```
String str = Double.toString(dValue);
double dValue = Double.valueOf(str);
```

String

```
String substr = str.substring(iBegin, iEnd);
String substr = str.substring(iBegin);

int indexMatch = str.indexOf(strFind);
int indexMatch = str.lastIndexOf(strFind);

int iLength = str.length();

boolean bMatch = str.equals(strCompare);
```

Arrays

```
int iLength = iArray.length;

ClassName[] classArray = new ClassName[iNbrOfClasses];
classArray[0] = new ClassName();
classArray[1] = new ClassName();

int[]  iArray = new int[nLocs];
int[] iArray = {0,1,2};

int[][] iMatrix = new int[nLocsFirst][];
iMatrix[0] = new int[nLocs0];
iMatrix[1] = new int[nLocs1];
```

For Each

```
int[] iArray;
for (int iElement : iArray) {
        System.out.println(iElement);
}
```

Array Copy

```
System.arraycopy(strSrc[], iSrcElement, strDest[], iDestElement, iLength);
```

Parameterizing Raw to Generic Type

```
Vector<String> vecName = new Vector<String>();
JList<String> listName = new JList<String>();
```

Vector

```
Vector<Class> vec = new Vector<Class>();  Class: String, Integer , etc
int iLocs = vec.size();
vec.removeAllElements();

vec.addElement(str);
String str = (String)vec.elementAt(iLocation);

vec.addElement(new Integer(iValue));
int iValue = ((Integer)(vec.elementAt(iLocation))).intValue();

ClassName c = new ClassName();
v.addElement(c);
ClassName c = (ClassName)(v.elementAt(iLocation));
```

Exceptions

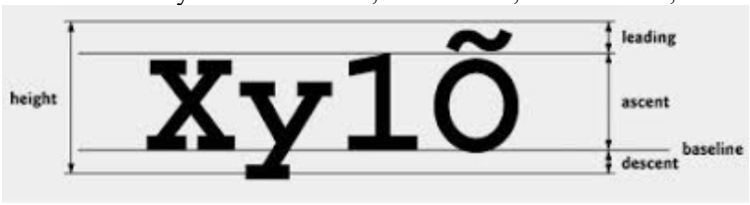
```
Throw(new Exception("Exception message."));
String strMessage = exc.getMessage();
```

WWW File Names

```
String strFileName = "https://fwesthoff.people.amherst.edu/filename.ext";
```

Font (java.awt.Font)

```
Font fontNew = new Font(strFontName, iFontStyle, iSize)
        strFontName = Font.DIALOG, Font.MONOSPACED, Font.SERIF, Font.SANS_SERIF, …
        iFontStyle = Font.PLAIN, Font.BOLD, Font.ITALIC, …
```



Unicode Characters

```
String strUnicode = FormattedOutput.getUnicode(0x00A9)      // 00A9 is the Unicode copyright
```

Colors

```
Color colorRed = Color.red;
Color colorNew = new Color(0xFF0000)      // FF0000 is the red hex spec
```

Detecting "Special" Keys

```
addMouseListener(new MouseAdapter() {
        public void mouse Released(MouseEvent e) {
                e.isControlDown(), e.isAltDown(), e.isMetaDown(), e.isShiftDown()
        }
}
```

Catching Key Strokes

```
public void jListPlayers_keyReleased(KeyEvent e) {
    int iKeyCode = e.getKeyCode();
    if(iKeyCode == KeyEvent.VK_UP || iKeyCode == KeyEvent.VK_DOWN) {

    }
}
```

JFrame

```
setAlwaysOnTop(boolean);
setUndocorated(boolean);
```

JDialog – Modal Mode

```
setModal(true);              // false: Prevent return until closed setVisible(false)
setVisible(true);
```

JList

```
JList<Class> list = new JList<Class>();                Class: String, Integer, …
list.setListData(Object[]);
list.setListData(Vector);
list.setListData(new Vector())                                  // clears list

list.setVisibleRowCount(iRows);

list.setSelectedIndex(iIndex);
list.setSelectedValue("Value"), true);                          // true allows scrolling
int iIndex = list.getSelectedIndex();
str strEntry = (String)list.getSelectedValue();                 // null if none selected

boolean3sum = (list.getModel()).getSize();
str strIndex = (String)(list.getModel()).getElementAt(iIndex);
```

JComboBox

```
JComboBox<Class> comboBox = new JComboBox<Class>();          Class: String, Integer, …

comboBox.addItem(iValue);
comboBox.setSelectedIndex(iIndex)
comboBox.getSelectedIndex()

comboBox.addItem(str);
comboBox.setSelectedItem(str);
str = (String) comboBox.getSelectedItem();
```

JComponent (JLabels, etc.)

```
component.setForeground(color);

component.setBackground(color);
component.setOpaque(true);          // This is necessary and can be done in the constructor

Border border = BorderFactory.createLineBorder(color[, iWidth]);      // Line border
Border border = BorderFactory.createEmptyBorder();                    // No border

component.setBorder(border);

// Rounded border corners
addComponentListener(new ComponentAdapter() {
        @Override
        public void componentResized(ComponentEvent e) {
                setShape(new RoundRectangle2D.Double(0, 0, getWidth(), getHeight(),
                        dArcWidth, dArcHeight)
        }
}
```

JCheckBox

itemStateChanged(): Captures a checkbox click (Others don't do so as effectively)

JProgressBar

```
private void startTask() {                          // Call by the action button or whatever
        // Set progress bar parameters
        progressBar.setMaximum(100);
        progressBar.setValue(0);
        progressBar.setStringPainted(boolean);      // true: Print; false: Print nothing
        TaskInnerClass task = new TaskInnerClass();
        task.start();
}
private void runTask() {
        progressBar.setValue(iIteration)
        progressBar.setString(str);                 // If null, prints percent complete
        // Do the work
}

class TaskInnerClass extends Thread {
        public void run() {
                progressBar.setVisible(true);
                runTask();
                progressBar.setVisible(false);
        }
}
```

JFileChooserFilters

```
fcc.setFileFilter(new SpecialFileFilter());

class MySpecialFileFilter extends javax.swing.filechooser.FileFilter {
    public String getDescription() {
        return "Special files(*.*)";
    }
    public boolean accept(File f) {
        String str = f.getName();
        if(str.indexOf("~") < 0) return true;
        return false;
    }
```

ButtonGroup

```
ButtonGroup bg = new ButtonGroup();
bg.add(jButton);
```

In Eclipse: Right click on button in the design window.

JScrollPane

Put component in the JScrollPane

JTextArea

```
setLineWrap(boolean b);
setWrapStyleWord(boolean b);
```

Focus

```
component.requestFocus();
component.requestFocusInWindow();
```

Abstract Classes

```
class CallingClass {
        ClassAbstract classAbstract;

        classAbstract = new ClassGeneric1(); or classAbstract = new ClassGeneric2();

        classAbstract.methodAbstract ()
}

public abstract class ClassAbstract {
        abstract xxxx methodAbstract();
}

public class ClassGeneric1 extends ClassAbstract {
        public xxxx methodAbstract() {
        }
}
public class ClassGeneric2 extends ClassAbstract {
        public xxxx methodAbstract() {
        }
}
```

Interfaces

```
CallingClass.java
    class CallingClass implements ImplementedClass Interface {
        // Calling class must include the methods specified in ImplementedClassInterface.java :
            public void requiredInterfaceMethod1(…) { … }
            public void requiredInterfaceMethod2(…){ … }

        // When created, the class ImplementedClass constructor must include "this" in its
        // argument list to connect CallingClass to ImplementedClass
            ImplementedClass implementedClass = new ImplementedClass (this, …);
                ⋮
    }

ImplementedClass.java
    class ImplementedClass {
        ImplementedClassInterface inter;

        ImplementedClass(ImplementedClassInterface inter, …) {
            this.inter = inter;                     // Save CallingClass location
                ⋮
        }

        inter.requiredInterfaceMethod1(…);          // Return to CallingClass
        inter.requiredInterfaceMethod2(…);          // Return to CallingClass
    }

ImplementedClassInterface.java
    public interface ImplementedClassInterface {
        public void requiredInterfaceMethod1(…);
        public void requiredInterfaceMethod2(…);
                ⋮
    }
```

Inner and Static Nested Classes

```
TestMainClass testMainClass = new TestMainClass();
TestMainClass.TestInnerClass testInnerClass = testMainClass.new TestInnerClass();
System.out.println(testInnerClass.subroutine());
System.out.println(TestMainClass.TestStaticNestedClass.subroutine());

public class TestMainClass {
	public class TestInnerClass {
		public String subroutine() {
			return "TestInnerClass: subroutine()";
		}

	}

	public static class TestStaticNestedClass {
		public static String subroutine() {
			return "TestStaticNestedClass: subroutine()";
		}
	}
}
```

Inner Class "Equilivalence"

```
method(…, new NewClass() {
	// NewClass code
});

method(…, new NewClass());
class NewClass {
	// NewClass code
}
```

addNotify()

Some operations cannot be completed in the constructor and an error results. To resolve this use the following code and implement the operations there (See CadView.java for implementation):

```
boolean bFirst = true;
public void addNotify() {
	super.addNotify();
	if (bFirst) {
		//viewGraph.setGraphDefaults();

		bFirst = false;
	}
}
```

Mouse Listener

```
gLabel.addMouseListener(new MouseAdapter() {
        public void mouseExited(MouseEvent event) {
                Insert code here;
        }

        public void mouseReleased(MouseEvent event) {
                Insert code here;
        }

        public void mousePressed(MouseEvent event) {
                Insert code here;
        }
});
```

Determine which mouse button: bRightButton = SwingUtilities.*isRightMouseButton*(event);

Mouse Motion Listener

```
gLabel.addMouseMotionListener(new MouseMotionAdapter() {
        public void mouseDragged(MouseEvent event) {
                Insert code here;
        }

        public void mouseMoved(MouseEvent event) {
                Insert code here;
        }
});
```

Mouse Event Information

Mouse button

```
bRightButton = SwingUtilities.isRightMouseButton(event);
```

Mouse location

```
iX = event.getX();
iY = event.getY();
```

Component Listerners

```
component.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {
                Insert code here;
        }
});

component.addFocusListener(new FocusAdapter() {
        public void focusGained(FocusEvent e) {
                Insert code here;
        }
});

component.addKeyListener(new KeyAdapter() {
        public void keyReleased(KeyEvent e) {
                Insert code here;
        }
});
```

List of Files

```
File f = new File(strDirectoryPath);
String[] strList = f.list();                    // Includes all files in directory
String[] strList = f.list(new MyFilter);        // Includes only "filtered" files

import java.io.FilenameFilter;
class MyFilter implements FilenameFilter {
        public boolean accept(File fDirectory, String strFilename) {
                if(…) return true;
                return false;
        }
}
```

Window Closing Action

```
setDefaultCloseOperation(…);
        EXIT_ON_CLOSE
        HIDE_ON_CLOSE
        DISPOSE_ON_CLOSE
        DO_NOTHING_ON_CLOSE

// Catch closing
setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
        ⋮
addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(java.awt.event.WindowEvent e) {
                Insert code here.
        }
});
```

Terminating a Process

```
System.exit(iErrorCode)                 // Typically, iErrorCode = 0.
```

Convert String to Array

```
strArray[] strArray = str.split(" ");
```

Pausing a Thread

```
Thread.sleep(lMilliSeconds);
```

Sort Array

Arrays.sort(obj[]): Sort obj[] in ascending order

```
Arrays.sort(int[] iArray)
Arrays.sort(double[] dArray)
Arrays.sort(String[strArray)
```

Arrays.sort(obj[], Collections.reverseOrder()): Sort obj[] in descending order

```
Arrays.sort(int[] iArray, Collections.reverseOrder())
Arrays.sort(double[] dArray, Collections.reverseOrder())
Arrays.sort(String[strArray, Collections.reverseOrder())
```

```
Arrays.sort(obj[], new NewCompator<Obj>() {
        public int compare(Obj obj0, Obj obj1){
                return obj0-obj1;  // i0-i1, Math.signum(d0-d1), str0.compareTo(str1)
        }

        public int getInteger() {
                return i;
        }
        public int getDouble() {
                return d;
        }
        public int getString() {
                return str;
        }
});

class MixedArray {          // Creates the mixed array
        int i;
        double d;
        String str;

        Sorter(int i, double d, String str) {
                this.i = i;
                this.d = d;
                this.str = str;
        }

        public int getInteger() {
                return i;
        }
        public int getDouble() {
                return d;
        }
        public int getString() {
                return str;
        }
}
```

System

System.out.print(str);
System.out.println(str);

System.getProperty(str)

java.version	Java Runtime Environment version
java.vendor	Java Runtime Environment vendor
java.vendor.url	Java vendor URL
java.home	Java installation directory
java.vm.specification.version	Java Virtual Machine specification version
java.vm.specification.vendor	Java Virtual Machine specification vendor
java.vm.specification.name	Java Virtual Machine specification name
java.vm.version	Java Virtual Machine implementation version
java.vm.vendor	Java Virtual Machine implementation vendor
java.vm.name	Java Virtual Machine implementation name
java.specification.version	Java Runtime Environment specification version
java.specification.vendor	Java Runtime Environment specification vendor
java.specification.name	Java Runtime Environment specification name
java.class.version	Java class format version number
java.class.path	Java class path
java.library.path	List of paths to search when loading libraries
java.io.tmpdir	Default temp file path
java.compiler	Name of JIT compiler to use
java.ext.dirs	Path of extension directory or directories
os.name	Operating system name
os.arch	Operating system architecture
os.version	Operating system version
file.separator	File separator ("/" on UNIX)
path.separator	Path separator (":" on UNIX)
line.separator	Line separator ("\n" on UNIX)
user.name	User's account name
user.home	User's home directory
user.dir	User's current working directory

Locating JRE on Mac

- /Macintosh HD/Library/Internet Plug-Ins
- Right click JaraAppletPlugin.plugin
- Click Show Package Contents
- Home includes the bin and lib directories

Java Control Panel on Mac

- Click on System Settings icon
- Click on the Java icon at the bottom of the left panel list
- Click on the Java tab
- Examine the Path (/Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/…
- In Finder, go to /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/
- Right click on JavaAppletPlugin.plugin
- Click on Show Package Contents
- Click on Home

Locating JRE on PC

C:\Program Files\Java

FW Library Documentation

(FW) ConvertString

```
String[] ConvertString.toString(str, strDelimiter);          // Use str.split(" ") instead
int[] ConvertString.toInteger(str, strDelimiter);
double[] ConvertString.toDouble(str, strDelimiter);
boolean[] ConvertString.toBoolean(str, strDelimiter);        // strDelimiter: add + to eliminate empty cells

double[] ConvertString.stripComment(str[]);
double ConvertString.stripComment(str);

String ConvertString.clean(str);
String ConvertString.substring(str, iBegin, iEnd)

String ConvertString.replaceAllVector(str[], strOld, strNew);
String ConvertString.replaceUnderscores(str, strOld, strNew);
```

(FW) ConvertStringToArray

```
String[] ConvertStringToArray.toString(strSource);
String[] ConvertStringToArray.toString(strSource, strDelimitor);
int[] ConvertStringToArray.toInteger(strSource);
int[] ConvertStringToArray.toInteger(strSource, strDelimitor);
double[] ConvertStringToArray.toDouble(strSource);
double[] ConvertStringToArray.toDouble(strSource, strDelimitor);
```

(FW) FormattedOutput

```
String FormattedOutput.encode(iValue [,strAllign, iLength]);
String FormattedOutput.encode(dValue, iLength);
String FormattedOutput.encode(dValue, strAllign, iLength, iDecimals);
String FormattedOutput.encode(iBlanks);
String FormattedOutput.insertCommas(str);
String FormattedOutput.encodeDollarsAndCents(dValue [,strAllign, iLength]);
```

(FW) JavaMail

```
JavaMail(strMailServer, strSender, strAddressee, strMessage) throws Exception
boolean isMailOK();
```

(FW) Clipboard

```
Clipboard.copyText(this, str);
String Clipboard.pasteText(this);
```

(FW) JWindowUtility

```
JWindowUtility.center(parentWindow, window);
JWindowUtility.center(window);
JWindowUtility.resizeForInsets(window);
boolean JWindowUtility.resizeComponent(jComponent, bWidth, bHeight);
String JWindowUtility.getMyDocumentsPath(strDirectory);
String JWindowUtility.getMyDocumentsPath();
String JWindowUtility.constructTitle(strModule, strVersion, strFilename);
String JWindowUtility.constructTitle(strModule, strVersion);
```

(FW) DateTimeAux

```
long lTime = DateTimeAux.getTime();         // Get current time
long lTime = DateTimeAux.getTime(lMilliSeconds);
long lTime = DateTimeAux.getTime(iYear, iMonth, iDay, iHour, iMinute, iSecond) ;

str = DateTimeAux.formatDate(lMilliSeconds);
str = DateTimeAux.formatDate(lMilliSeconds, strSeparator);
str = DateTimeAux.formatTime(lMilliSeconds);
str = DateTimeAux.formatTime(lMilliSeconds, strSeparator);

int[] = DateTimeAux.getDate(lMilliSeconds);
int[] = DateTimeAux.getDateAndTime(lMilliSeconds);

int = DateTimeAux.getDayOfWeek(lMilliSeconds);
```

(FW) Sorters

```
void Sorters.sortArray(strArray);
void Sorters.sortArray(iArray);
void Sorters.sortArray(lArray);
void Sorters.sortArray(dArray);

void Sorters.sortArray(iLength, strArray);
void Sorters.sortArray(iLength, iArray);
void Sorters.sortArray(iLength, lArray);
void Sorters.sortArray(iLength, dArray);

int[] Sorters.sortArrayIndices(strArray);
int[] Sorters.sortArrayIndices(iArray);
int[] Sorters.sortArrayIndices(lArray);
int[] Sorters.sortArrayIndices(dArray);

int[] Sorters.sortArrayIndices(iLength, strArray);
int[] Sorters.sortArrayIndices(iLength, iArray);
int[] Sorters.sortArrayIndices(iLength, lArray);
int[] Sorters.sortArrayIndices(iLength, dArray);
```

(FW) JOptionBox

JOptionBox

```
public static int optionBox(String strTitle, String strMessage, String strOptionsString, int
    iDefaultOption)
        // Center box in screen
public static int optionBox(JFrame frame, String strTitle, String strMessage, String
    strOptionsString, int iDefaultOption)
        // Center box in frame
```

(FW) JQueryBox

```
JQueryBox qb = new JQueryBox(strTitle, strMessage, strLabel);
JQueryBox qb = new JQueryBox(strTitle, strMessage, strLabelArray);
qb.isButton(strLabel);
qb.dispose();
```

(FW) ColorAux

```
white black gray darkGray lightGray
red darkRed lightRed paleRed
green darkGreen mediumGreen lightGreen paleGreen
blue darkBlue lightBlue paleBlue
magenta purple cyan pink darkPink orange darkOrange yellow darkYellow turquoise brown
```

(FW) Copy a File from the Web

```
JWebCopyUtility webCopy = new JWebCopyUtility(null, progressBar);
webCopy(strWebPath, strPCPath, strFileNameExt);
webCopy.setVisible(true);
```

(FW) Directory History

```
JDirectoryHistory dh = new JDirectoryHistory(strDirectoryHistoryFilename);
    For example, JDirectoryHistory dh = new JDirectoryHistory("Econ55);
    This is a text file that is in the My documents directory My Histories
DirectoryHistory dh = new DirectoryHistory(strDirectoryHistoryFilename);
String strLastDirectory = dh.getLastDirectory();
    strLastDirectory ends with a back slash, \.
dh.saveLastDirectory(jchooser);
     public void this_windowClosing(WindowEvent e) {
          dh.saveLastDirectory(jchooser);
     }
dh.saveLastDirectory(strFileDirectory);
```

(FW) JTextPaneOutput

```
JTextPaneOutput textPaneOutput = new JTextPaneOutput(jScrollPane);

textPaneOutput.setText(str);
textPaneOutput.append(str);

textPaneOutput.clearAttributes();

textPaneOutput.setForegroundColor(c);

textPaneOutput.setBold(bool);
textPaneOutput.setItalics(bool);
textPaneOutput.setUnderline(bool);

textPaneOutput.setFontFamily(str);                  // Serif, Sansserif, Monospace, Dialog
textPaneOutput.setFontSize(int);

textPaneOutput.setSubscript(bool);
textPaneOutput.setSuperscript(bool)

textPaneOutput.setTab(int);
textPaneOutput.setTabs(int[]);
textPaneOutput.setTabs(int[], strAlign[]);          // strAlign "l", "r", "c", "d"

textPaneOutput.setLeftIndent(int)
textPaneOutput.setAlignment(strAlign);              // strAlign "l", "r", "c", "d"
```

(FW) Split

```
String[] Split.toString(str, strDelimiter)          // Needed for Windows File.separator issue
int[] Split.toInteger(str, strDelimiter)
double[] Split.toDouble(str, strDelimiter)
boolean[] Split.toBoolean(str, strDelimiter)
```

(FW) JGraph

```
JGraph graph = new JGraph(label, strAxisRange, strAxisNames, strAxisNumericalLabels);
        strAxisRange = XMin YMin XMax YMax [XLabelDisplayFactor YLabelDisplayFactor]
        strAxisNames = "XName YName";
        strAxisNumericalLabels = " T/F T/F";
JGraph graph = new JGraph(label, strAxisRange);
drawAxes();

setPointDiameter(iPointDiameter)   // 8 is a good value
setPointColor(color);
drawPoint(dX, dY, iPointDiameter, color, strSpec);
        strSpec
            X: draw line from point to x-axis
            Y: draw line from point to y-axis
            R: draw line from point to the extreme right of graph
            U: draw line from point the top of graph
drawPoint(dX, dY [, iPointDiameter, color]);
drawPoint(dXYs[] [, iPointDiameter, color]);

setLineWidth(iLineWidth)            // 1 is the default
drawLine(dX0, dY0, dX1, dY1 [, iLineWidth, color]);
drawLine(dX, dY, dSlope [, iLineWidth, color]);
drawLine(dXYs[][] [, iPointDiameter, color]);
drawLineEquation(dCoefX, dCoefY, dConst [, iLineWidth, color]);

drawPolygon(dXs[], dYs[] [, iLineWidth, color]);
fillPolygon(dXs[], dYs[] [, color]);

setFontName(strFontName);
setFontColor(color);
setFontSize(iFontSize);
drawString(str, dX, dY, cPosition [, iFontSize, color]);
        cPosition
            U or ^: above and center horizontally
            D or _: below and center horizontally
            L or <: left and center vertically
            R or >: right and center vertically
            blank: right and no centering
getStringHeight(str);
getStringWidth(str);

double[] getMouseXYCoordinates();
        Use WindowBuilder to add a motion listener to graph label;
        then add call to getMouseXYCoordinates() to get mouse coordinates:
                label.addMouseMotionListener(new MouseMotionAdapter() {
                        @Override
                        public void mouseXXXX(MouseEvent e) {  // XXXX: Dragged, Pressed, …
                                dMouseXYCoordinates = graph.getMouseXYCoordinates();
                        }
                });
```

(FW) Draw Curve Example

```
private void DrawCurveXXXX {
        :
        :
    DrawCurveXXXX drawCurveXXXX = new XXXXDrawCurve(graph, …);
    drawCurveXXXX.drawCurve(double dXMinimum, double dXMaximum, …) {
        :
        :
}

class DrawCurveXXXX implements JGraphCurveInterface {        // MainClass that draws graph
    JGraph graph;
    DrawCurveXXXX(JGraph graph, …); {
        this.graph = graph;     // Store graph
            :
            :
    }

    public double calCurveYFromX(double dX) {                 // Interface routine
        // Calculate dY;
            :
            :
        return dY;
    }

    public drawCurve(double dXMinimum, double dXMaximum, …) {
        graph.drawCurve(this, dXMinimum, dXMaximum [, iLineWidth, color]);
        graph.drawCurveAndLabel(this, dXMinimum, dXMaximum, strLabel [, iLineWidth, color]);
    }
}
```

(FW) JWebCopyUtility

```
    JWebCopyUtility(progressbar)
    copyFile(strUrlPath, strPCPath)
```

(FW) JDisappearingMessage

JDisappearingMessage(frame, lDuration, strMessage);

(FW) FileUtility

boolean FileUtility.isOSMac()
boolean FileUtility.isOSWindows()
boolean FileUtility.isOSUnix()

String FileUtility.constructDocFilePath(strDocDirectory, strFileName, strFileExtension)
String FileUtility.constructDocFilePath(strDocDirectory, strFileNameExtension)
String FileUtility.constructDocFilePath(strDocFilePath)

String FileUtility.constructFilePath(strDirectory, strFileName, strFileExtension)
String FileUtility.constructFilePath(strDirectory, strFileNameExtension)
String FileUtility.constructFilePath(strFilePath)

String FileUtility.webPath(strHttpDir, strFileName, strFileExt)
String FileUtility.webPath(strHttpDir, strFileNameExt)
String FileUtility.webPath(strHttpDirFileNameExt)

String[] FileUtility.getFileDirectoryNameExtension(strFilePath)
String FileUtility.getFileDirectory(strFilePath)
String FileUtility.getFileName(strFilePath)
String FileUtility.getFileExtension(strFilePath)

boolean FileUtility.isFile(strFilePath)
boolean FileUtility.isDirectory(strDirectoryPath)

boolean FileUtility.renameFile(strFilePathInput, strFilePathOutput)
boolean FileUtility.deleteFile(strFilePath)
void FileUtility.copyFile(strFilePathInput, strFilePathOutput) throws Exception
void FileUtility.copyFile(strFilePathInput, strFilePathOutput, lBufferMax) throws Exception

boolean FileUtility.createDirectory(strDirectory)
boolean FileUtility.removeDirectoryTree(strDirectory)
boolean FileUtility.removeDirectory(strDirectory)

long FileUtility.lastModified(strFilePath)
void FileUtility.setLastModified(strFilePath, lDate)

String[] FileUtility.getListOfFiles(strDirectory)
String[] FileUtility.getListOfFiles(strDirectory, strExtension)
String[] FileUtility.getListOfDirectories(strDirectory)

String FileUtility.stripAndAppendSeparator(str)

String FileUtility.correctFileSeparator(strFilePath) // Corrects all Mac/Windows separator errors
String FileUtility.condenseDuplicateSeparators(strFilePath)
String FileUtility.abbreviateFilePathName(strFilePathName)

(FW) Ascii File Read

```
try {
        FileAsciiRead far = new FileAsciiRead(strFilePath);

        String str;
        while( (str = far.readLine()) != null) {
                // process
        }
        far.close();
}
catch (Exception e) {
        String str = e.toString() ;
}
```

(FW) Ascii File Write

```
try {
        FileAsciiWrite faw = new FileAsciiWrite (strFileName);

        faw.println(str);
        faw.flush();
        faw.close();
}
```

(FW) Object File Read

```
try {
        FileObjectRead for = new FileObjectRead(strFileName);

        double[] dVector = (double[])for.readObject();
        String[][] strArray = (String[][])fori.readObject();
}
```

(FW) Object File Write

```
try {
        FileObjectWrite fow = new FileObjectWrite(strFileName);

        double[] dVector;
        fow.writeObject(dVector);

        String[][] strArray;
        fow.writeObject(String[][]);

        fow.flush();
        fow.close();
}
```

(FW) Random Access File Read (Applications Only)

```
try {
        FileRandomRead frr = new FileRandomRead(strFileName);

        Long lPointer = frr.getFilePointer();
        frr.seek(lPointer);

        int iValue = frr.readInt();
        double dValue = frr.readDouble();

        byte[] bArray = new byte[nLocs];
        frr.readFully(bArray);

        String str = frr.readUTF();
        boolean bValue = frr.readBoolean();
}
```

(FW) Random Access File Write (Applications Only)

```
try {
        FileRandomReadWrite frrw = new FileRandomReadWrite(strFileName);

        Long lPointer = frrw.getFilePointer();
        frrw.seek(lPointer);

        frrw.writeInt(iValue);
        frrw.writeDouble(dValue);

        frrw.write(bArray);

        frrw.writeUTF(str)
        frrw.writeBoolean(bValue);
}
```

(FW) JarInstallationUtility

```
checkForUpdate(frame)
getAuxDirFilePath(strFileNameExt)
```

(FW) JFileSelector.java [class JFileSelector extends JFrame implements JTableDirectoryInterface]

```
public class XXXX extends JFrame implements JFileSelectorInterface {
    public void selectedFilePaths(strFilePaths[]) { … }     // Required interface routine

    JFileSelector fileSelector = JFileSelector(inter, frame, strMode, strFileExtension);
        // Typically, inter: this and frame: this
        strMode: O[pen] S[ave] M[ultiple] D[irectory]
        strFileExtension: null for all files

    void setDirectory(strDirectory)
    void setFileNameExt(strFileNameExt)

    void setBaseDirectory(strBaseDirectory)                 // Default: Documents
    void setPartialDirectory(strPartialDirectory)

    boolean isBaseDirectory()                               // true if Documents
```

(FW) JTableDirectory [class JTableDirectory implements JTableAuxInterface]

```
public class XXXX extends JFrame implements JTableDirectoryInterface {
    public void cellSelected(String strDirecoryPath, String[] strMultipleFilePaths) {
        // Required interface routine

    JTableDirectory tableDirectory = JTableDirectory(inter, scrollPaneFile, comboBoxBaseDirectory,
            textFieldDirPath, strFileExtension);
        // Typically, inter: this
        strFileExtension: null for all files
        textFieldDirPath, strFileExtension);

    void setPartialDirectory(strPartialDirectory)
    void setMultipleSelection(bMultipleSelection)                   // Default: (false)
    void setShow(bDirectories, bFiles)
    void setArrayColors(colorDirectories[2], colorDirectories[2])

    String getBaseDirectory()
    String getPartialDirectory()
    String getFullDirectory()

    void fillTable()
    void clearSelection();
```

(FW) JTableAux [class JTableAux extends JTable]

```
public class XXXX extends JFrame implements JTableAuxInterface {
    public void cellClicked() { … }      // Required interface routine

    JTableAux tableaux = JTableAux(inter, scrollPane, strArray[][]);
        // Typically, inter: this

    Setters
        void setVisible(bVisible);                                  // Default: true
        void setAutoResize(bAutoResize);                            // Default: false

        void setColumnWidthsString(strColumnWidths)                 // Default: All 100

        void setHeaderNamesString(strHeaderName)                    // Default: No header (null)

        void setHeaderColumnAlignsString(strAlign)                  // Default: All left
        void setArrayColumnAlignsString(strArrayAlign)              // Default: All left
            strAlign = L[eft] C[enter] R[ight]

        void setHeaderFont(fontHeader)                              // Default: (Font.DIALOG, Font.PLAIN, 12)
        void setArrayFont(fontArray)                        // Default: (Font.DIALOG, Font.PLAIN, 12)
            fontHeader = new Font(strFontName, iFontStyle, iFontSize)
            strFontName = Font.DIALOG, Font.MONOSPACED, Font.SERIF, Font.SANS_SERIF, …
            iFontStyle = Font.PLAIN, Font.BOLD, Font.ITALIC, …

        void setBorderColors(colorTableBorders, colorGridBorders)           // Default: Black Black
        void setHeaderColors(colorHeaderBackground, colorHeaderForeground)  // Default: White Black
        void setArrayColors(colorBackground, colorForeground, iRow, iColumn)// Default: White Black
        void setArrayColorsAll(colorBackground, colorForeground, iRow, iColumn)

        void setSelectionColors(Color colorBackground, Color colorForeground)   // Default: Black White
        void setEditableColors(colorBackground, colorForeground)                // Default: LightGray White

        void setSelectionMode(String strSelection)                  // Default: None
            strSelection = “” “R[ow]” “Co[lumn]” “Ce[ll]”
        void setSelectionMultiple(boolean bMultipleSelection)       // Default: false

        void setEditableCell(iRow, iColumn, bEditable)              // Default: false

    Getters
        int[] getSelectedCell()         // If selection mode is Row, Column, or Cell

        int[] getSelectedRows()         // If selection mode is row
        int[] getSelectedColumns()      // If selection mode is column

    Flush: Implements the last cell edit into strArray
        void flush()                    // “Flushes” last edited cell into array
```

(FW) Html Utilities

String HtmlCodeUtility.tableToString(String str);
String HtmlCodeUtility.trimHtmlTable(String strTable);

String HtmlCodeUtility.getHtmlLines(FileAsciiRead far, String strStart, String strStop);
strStart = “FirstStartString`SecondStartString`_”); _ indicates a line skip
strEnd = “FirstStopString`SecondStopString`_”); _ indicates a line skip
strEnd = “”; return immediately
NB: If only one start and stop string no grave accenst are needed.

String HtmlCodeUtility.stripHtmlCode(String str);

(FW) FixedPointsIntegerDialog

class MyParentFrame implements FixedPointsIntegerInterface

```
double[] dPointFrom, double[] dPointTo;
FixedPointsIntegerDialog fpd = new FixedPointsIntegerDialog(
    this, double[] dPointFrom, double[] dPointTo);
//fpd.setStopAfterMajorIteration(b); // The default is false

// Start and Stop
fpd.setVisible(true);
fpd.dispose();

// Interface routines:
public void getMappedPoint() {
    ⋮
}
public String getSummaryText() {
    return null; for the default summary
    ⋮
```

(FW) SolveForRoot

```
class SolveForRootParent implements SolveForRootInterface {

    private double solve() {
        SolveForRoot solveForRoot = new SolveForRoot(this);
        :
        double dRoot = solveForRoot.interate(dXLowerBound, dXUpperBound, dTolerane);
        if(Double.isNaN(dRoot)) {
            // No root within bounds
        }
        return dRoot;
    }

    // Required interface routines:
    public double calculateYFromX (double dX, …) {
        :
        return dY;
    }
}
```

(FW) SimulationThread

```
    class SimulationParent implements SimulationThreadInterface {
```

Create the following:

- Start/Next button: Add ActionPerformed event handler
- Stop button: Add ActionPerformed event handler
- Pause checkbox
- Repetitions label

```
    private void startSimulation() {
        if(simulationThread == null) {
            simulationThread = new SimulationThread(this,
                buttonStart, buttonStop, checkBoxPause);
            //simulationThread = new SimulationThread(this,
            //          buttonStart, buttonStop, checkBoxPause);
        }
        simulationThread.startSimulation();
    }

    private void stopSimulation() {
        simulationThread.stopSimulation();
        simulationThread = null;
    }

    // Required interface routines:
    public void runRep () {
        :
    }
    public void reportResults(int iReps) {
        :
    }
}
```

(FW) SimulationThread Example

```
class LabCardDrawView extends JFrame {
    private static final long serialVersionUID = 1L;

    // Generic globals:
    LabCardDrawController c;
    LabCardDrawProcessSpecs processLabAndProbSpecs;

    // Graph globals
    JLabel labelHistogram;
    String strAxisDataRange;
    String strAxisNames;
    JGraphHist hist;

    /* Here are the keys:
       hist must be null at the start of a simulation so that it is initialized.
       After hist.graphHistogram(iCounts) a super.repaint() must be called.
    */

    public void graphResults() {
        // Get cout information
        int[] iCounts = c.getiCounts();

        // Initialize graphs
        if(hist == null) {
            try {
                hist = new JGraphHist(labelHistogram, 1, 2);
            }
                return;
            }
        }

        // Set number of bars
        hist.setNumberOfBars(iCounts.length);

        // Histogram
        hist.setHeader(strHistHeader);
        int iNumberIncrement = 1;
        for (int i = 0; i < iCounts.length; i = i + iNumberIncrement) {
            String str = Integer.toString(i + iSelectedCardValueMin);
            hist.setFooter(str, i, 0);
        }
        hist.graphHistogram(iCounts);
        super.repaint();
    }

    public void startSimulation() {
        if(!c.isbKeepThreadRunning()) {
            buttonStop.setVisible(true);
        }
        hist = null;
        buttonStart.setVisible(checkBoxPause.isSelected());
        c.startSim(checkBoxPause.isSelected());
    }
}
```

(FW) JFileChooserCheckApplication[1]

```
JFileChooserCheckApplication fcca = new JFileChooserCheckApplication();
JFileChooserCheckApplication fcca = new JFileChooserCheckApplication(strExtension);
JFileChooserCheckApplication fcca = new JFileChooserCheckApplication(strExtensionArray);

String fcca.getFilePath(bMustExist);          // Null if no file selected
        Must be called to activate.

fcca.setMultiSelection(b);
String[] = fcca.getFilePaths(b);              // String.length = 0 if no file selected

String fcca.getFileName();
String fcca.getTitle(strTitle);
boolean fcca.isApplication();
```

(FW) JFileChooserCheckApplet

```
JFileChooserCheckApplet fcca =
        new JFileChooserCheckApplet(strWebLocation, strDirectoryFileName);
                strWebLocation = http://....
                strDirectoryFileName:
                        Description for File 1  File1Name.Ext
                        Description for File 2  File2Name.Ext

String fcca.getFilePath(bMustExist);          // Null if no file selected
        Must be called to activate.

String fcca.getFileName();
String fcca.getTitle(strTitle);
boolean fcca.isApplication();
```

Using (FW) JFileChooserCheckInterface for Applications or Applets

```
facca = new JFileChooserCheckApplication(…) or JFilechooserCheckApplet(…)
MyClass myClass = MyClass(facc)
```

---

[1] **Swing: (FW) JFileChooserCheck (Used by JFileChooserApplication)**

```
JFileChooserCheck fcc = new JFileChooserCheck();
JFileChooserCheck fcc = new JFileChooserCheck(bMustExist);
JFileChooserCheck fcc = new JFileChooserCheck(bMustExist, strExtension);
JFileChooserCheck fcc = new JFileChooserCheck(bMustExist, strExtensionArray);

fcc.setTitle(strTitle);
fcc.setFileExtension(strExtension);
fcc.setFileExtension(strExtensionArray);
fcc.setFileMustExist(bMustExist);
fcc.setVisible(true/false);

File fcc.getFile(); // Null if no file selected
        Must be called.

String fcc.getFilePath();
String fcc.getFileDirectory();
String fcc.getFileName();
String fcc.getFileExtension();
```

(FW) JSpreadSheet

Column and Row Notation

strColumnNames[0]	strColumnNames[1]	strColumnNames[2]	…
strArray[0][0]	strArray[0][1]	strArray[0][2]	…
strArray[1][0]	strArray[1][1]	strArray[1][2]	…
strArray[2][0]	strArray[2][1]	strArray[2][2]	…

If there are row names, the first strArray column, strArray[i][0], contains the names.

Class MyParentFrame implements JSpreadSheetInterface

First, create a JScrollPane.

Constructors: Two types – one for separate column names and array and one for entire spreadsheet

```
JSpreadSheet ss =
    // Separate column names and data
    JSpreadSheet(this, JScrollPane scrollPane, String[][] strArray, String[] strColumnNames,
        boolean bLockFirstColumn, boolean[][] bCellEditable)
    JSpreadSheet(this, JScrollPane scrollPane, String[][] strArray, String[] strColumnNames,
        boolean bLockFirstColumn)
        // NB: All strArray cells will be editable.
    JSpreadSheet(this,  JScrollPane scrollPane, String[][] strArray, String[] strColumnNames)
        // NB: All strArray cells will be editable and first column is locked.

    JSpreadSheet(this, JScrollPane scrollPane, String[][] strEntireSpreadSheet,
        boolean bLockFirstColumn,  boolean[][] bCellEditable)
    JSpreadSheet(this, JScrollPane scrollPane, String[][] strEntireSpreadSheet,
        boolean bLockFirstColumn)
        // NB: All strArray cells will be editable.
    JSpreadSheet(this,  JScrollPane scrollPane, String[][] strEntireSpreadSheet)
        // NB: All strArray cells will be editable and first column is locked.

    // Interface routine:
    public boolean isCellChangeValid(int iRow, int iColumn, String str) {
        // When appropriate, use ss.isDouble() to check for valid entry.
        ⋮
    }

    // Auxillary routines:
    ss.flush();                 // Called to flush out last spread sheet entry
    ss.isDouble();              // Used by isCellChangeValid to check for valid entry
    ss.getDouble();
    ss.clearSpreadSheet();
```

(FW) FileManagement

class MyParentFrame implements JFileManagementInterface

Create the following:

- File JMenuBar, JMenu, JMenuItem
- The following menu items: New, Open, Save, SaveAs, and Close

```
MyFileClass myFileClass = new MyFileClass(strFileExtension, jMenuItemSave,
        jMenuItemSaveAs, jMenuItemClose)

// Menubar action routines
myFileClass.newFile();
myFileClass.openFile();
myFileClass.saveFile();
myFileClass.saveAsFile();
myFileClass.closeFile();
```

```
class MyFileClass extends JFileManagement implements JFileManagementInterface() {
    MyFileClass(String strFileExtension, JMenuItem jMenuItemSave,
        JMenuItem jMenuItemSaveAs, JMenuItem jMenuItemClose)

        // NB: One of the following two constructors is necessary
        super(strFileExtension, jMenuItemSaveAs, jMenuItemSave, jMenuItemClose);
        super();            // I don't understand this one, unless the other items are unwanted.

        // NB: The following call is necessary to set the interface class
        super.setInterface(this);

        NB: Typically, you want to pass the data entry class (e.g., the class that contains the
        spread sheet to that this class can have access to the data.

        // JFileManagement auxillary routines:
        super.getFilePath();
        super.getFileName();
        super.changes();  // Informs JFileManagement that changes have occurred

        // Interface routines:
        public void initializeFile() {          // Sets up and initializes the data classes
            ⋮
        }
        public void readFile() {
            ⋮
        }
        public void writeFile() {
            ⋮
        }
        public void clearFile() {               // Don't know what this does
            ⋮
        }

        // NB: bChanges must be set to true if any data are changed.
        public boolean isChanged() {        // Don't know how this works
            return bChanges;
        }
```

JEconLab (fweconlabswing.jar)

(FW) JEconLabSlider

JEconLabSlider slider = new JEconLabSlider(scrollbar [, label, strPrefix])

init(strSpecs) // strSpecs = “minimum maximum iteration default”
init(strSpecs, strPrefix) // strSpecs = “minimum maximum iteration default”
init(label)
init(label, strPrefix)

setSliderMinimum(strMinimum)
setSliderMaximum (strMaximum)
setSliderIncrement (strIncrement)
setSliderDefault (strDefault)
setSliderDefault ()

setSliderPrefix(strPrefix)
setSliderDecimals(iDecimals)

setSliderVisible(bVisible)
setSliderScrollbarVisible(bVisible)
setSliderLabelVisible(bVisible)
setSliderVisible(strVisible)
- B: Both scrollbar and label visible
- S: Scrollbar only visible
- L: Label only visible
- N: Neither visible

setSliderValue(dValue)
updateSliderValue()

calDecimalPlaces(str)
calDecimalPlaces(strVector[])

getSliderInteger()
getSliderDouble()
getSliderString()

getSliderMinimum()
getSliderMaximum()
getSliderIncrement()
getSliderDefault()
getSliderDecimals()

isSliderValueEqual(dValue)
isSliderVisible()

(FW) JEconLabSlider

Nontoggle Commands

`CLR – Clears all formatting

`WHT – White
`BLK – Black
`DGY – Dark gray
`GRY – Gray
`LGY – Light gray
`DRD – Dark red
`RED – Red
`LRD –Light red
`GRN – Green
`MGR – Medium green
`DGR – Dark green
`DBL – Dark blue
`BLU – Blue
`LBL – Light blue
`MAG – Magenta
`PUR – Purple
`CYN – Cyan
`DTR – Dark tuquoise
`TUR – Turquoise
`PNK – Pink
`DOR – Dark organge
`DYE – Dark yellow
`YEL – Yellow
`BRN – Brown

Toggle Commands

`BLD – Bold
`ITL – Italics
`UND – Underline
`SUB –Subscript
`SUP – Superscript
`IND – Indent

Special Characters

`0x2022 – Bullet
`0x25E6 – White bullet
`0x2212 – Minus sign
`0x0381 – Alpha
`0x0382 – Beta
`0x03B3 – Gamma
`0x0394 – Delta upper case
`0x00A2 – Cents
`0x2190 – Left arrow
`0x2191 – Up arrow
`0x2192 – Right arrow
`0x2193 – Down arrow
`0x21D0 – Left implies arrow
`0x21D2 – Right implies arrow

(FW) JEconLabSearch

JEconLabSearch search = new JEconLabSearch(sliderSearch
[, labelSearch, strSearchMessages])

init(strSearchMessages)
init(strSliderSpecs)

isSliderValueOptimal(dOptimalValue)
isSliderValueZero(dValue, dTolerance)

Example

```
String strMessages = getMessages();
econLabSearch = new JEconLabSearch(econLabScrollBar,
                    labelMessages, strMessages);
String strEconScrollBarSpecs = getEconScrollBarSpecs();
econLabScrollBar.init(strEconScrollBarSpecs);

dOptimalValue = setOptimalValue();

String strEconScrollBarSpecs = getEconScrollBarSpecs();
econLabScrollBar.init(strEconScrollBarSpecs);
String strMessages = getMessages();
econLabSearch.init(strMessages);
dOptimalValue = setOptimalValue();
```

(FW) JEconResizeLabPanel – Static routine

JEconResizeLabPanel.resizeLabPanel(frame, rectangle);

(FW) JEconLabProbSpecsTuple – Static routines

See LabTEMPLATEPane.java

(FW) EconLabButtonGroup

EconLabButtonGroup econLabButtonGroup = new EconLabButtonGroup (buttonGroup);

updateButtonGroup(buttonSelected);

(FW) EconLabElasticityUtility – Static routines

EconLabElasticityUtility.getPrecentChange(dInitial, dNew);

double[] EconLabElasticityUtility.convertElasticityToFunctionConstantAndSlope(
dElasticity, dQuantity, dPrice);
double[] EconLabElasticityUtility.convertElasticityToCurveInteceptAndSlope(
dElasticity, dQuantity, dPrice);

(FW) EconLabFunctionAndCurveUtility – Static routines

double[] EconLabDemandFunctionAndCurveUtility.calDemandCurveInterceptAndSlope(
dDemandFunctionConstant, dDemandFunctionOwnPriceCoef
[, dDemandFunctionCrossPriceCoef]);

double[] EconLabDemandFunctionAndCurveUtility.calDemandFunctionConstantAndCoef(
dDemandCurveIntercept, dDemandCurveSlope)

Eclipse-Window Builder Help

Creating a project:

- Toolbar Menu: New icon's **drop down list** (not the icon itself) – Java Project
- Specify the name of the project.
- Finish

Creating a frame, or applet, or …

- Highlight project.
- Toolbar Menu: New icon **itself** (not the icon's drop down list) – WindowBuilder – Swing Designer – JFrame (for application) or JApplet
- Name frame or applet
- Specify package (if necessary) and name
- Finish
- Add contentPane.setLayout(null) to frame constructor
- Delete import java.awt.BorderLayout;
- Right click warning icon – Quick Fix – Add default serial ID

Creating a panel

- Apparently the WindowBuilder Designer does not work properly until one component has been attached to the contentPane "by hand." So, use the JPanelTemplate.java in the subdirectory eclipse\Templates which includes btnDummy which can be deleted later.

Displaying panels

- Window (menu item) - Show View (drop down box)

Accessing WindowBuilder

- Right click Java file
- Open Java file - Open With – WindowBuilder Editor

Create a library Jar file and the Jar description file

- **File** (menu item): Select **Export**
- **Export window**: Select Java – JAR file
- **JAR File Specification** window
    - Select the resources to export window: Select files to include in the library
    - Select the export destination: Specify Jar file name. (I'm using the root directory of FWLibraries for the location of the library files; also, check the export name.)
    - Next
- **JAR Packaging Options**
    - Be certain the Save the description of this JAR file in the workspace is selected
    - Name Description file
- **Finish**

Create a library Jar file from a Jar description file

- Left double click Jar description file (*.jardesc)
- Finish

Add Libraries to Project

- Right click on project – Build Path – Configure Build Path…
- **Properties for *project*:** Click **Add Jars**
- **Jar Selection** window: Find and then click on the wanted library jar
  The library will now be listed as part of the project.

Cleaning Up Launch Configurations

- **Run** (menu item): Select **Run Configurations**
- Expand the **Java Application** line item in the left panel of the pop up window.
- Select the projects whose launch configuations are to be deleted, then press the **delete** key.
- **NB**: To create a new launch configuration for an application, run the application. On a Mac, this launch configuration will appear at the bottom of the list of launch configuations when creating a new runnable jar file. See below.

Create a Runnable Jar Application File and an Ant file: Produces a jar file that can be run as an application or applet and an ANT file which can be used in the future to create the jar file.

- **File** (menu item): Select **Export**
- **Export window**: Select **Runnable JAR file**
- **Runnable JAR File Export** window: **NB:** Be careful to specify the correct **Launch configuration**, **Export destination**, and **ANT script location**.
    - **Launch configuration:** Select.
        - **Special Mac Advice**: On a Mac, the launch configuration list is chaotic. Finding the correct configuration is difficult. See the section above to create a run configuration that will appear at the botton of the list of run configuations.
    - **Export destination:** Click Browse
    - **Library handling:** Select "Extract required …"
    - **Save as ANT script:** Check
    - **ANT script location:** Click Browse

  NB: As mentioned above, it is critical to specify the correct **Launch configuration**, **Export destination**, and **ANT script location**. **Check them!**

Create a Runnable Jar Application File Ant File: Produces a jar file that can be run as an application or applet

- In Eclipse, Right click the ANT file which has an xml extension.
- Click Run As
- Click 1 Ant Build
- There has been a problem with this which is easy to fix. See below

Create Runnable Jar Application Ant File "Bug"

- In Eclipse, Right click the ANT file which has an xml extension.
- Click Run As
- External Tools Configuration …
- Click JRE tab
- Select Run in the same JRE as the workspace

Modifying Xml Ant Scripts

- Xml ant scripts include both an invisible and a visible section:
    - Invisible section: Unique to the operating system (Windows of Mac) and the project. Therefore, the ant script must be created within Eclipse as described above initially.
    - Visible section: Can be edited in the standard way. For example, multiple jar files can be created within a single section by adding the <jar> … </jar> section.

Removing Color Highlights from Java Code

- Click the double X icon lying on the right of the first line lying on the right of the Conole panel.

Creating a Working Set

- Click the three vertically aligned circles in the top line of the Project Explorer
- Click Select Working Set… (third line from the top)
- Click New
- Enter name and specify projects

Adding a Project to a Working Sets

- Click the three vertically aligned circles in the top line of the Project Explorer
- Click Select Working Set… (third line from the top) and check the working set that is the destination.
- Click Edit
- The projects in the destination working space are displayed in the right panel. Move project.

Package Explorer Refreshing

The Package Explorer directory listing sometimes fails to list jar and xml files. To refresh the listing select the package and then press the F5 key.

WindowBuilder Parser Code

```
/**
 * @wbp.parser.constructor
 */
```

Installing Eclipse

- If necessary, download and install the Eclipse Installer: https://www.eclipse.org/downloads/packages/installer.
- Launch Eclipse Installer.
- Download and install Eclipse IDE for Java Developers. One the Mac Eclipse is accessed at MacHD/Users/fwesthoff/java-2020-06.

Installing Window Builder

- In Eclipse, click Help
- Click Eclipse Marketplace…
- Search for WindowBuilder
- Download WindowBuilder
- NB: It takes Eclipse a while to install WindowBuilder
- Exit and relaunch Eclipse

Autocompletion Tool

- Go to Advanced Content Assist window:
  - Mac: Eclipse > Preferences > Java > Editor > Content Assist > Advanced
  - Windows: Editor > Content Assist > Advanced
- Check Java Proposals in both windows
- Click Apply and Close
- You may have to close and reopen Eclipse or perhaps even restart computer for the settings to take effect.

Creating an Eclipse Project from a Standard Folder

- Select the folder (Not certain this is necessary, but it does not hurt)
- In Eclipse, click Java Project from the drop down menu
- Enter the folder name and click the Finish button
- The project has been added to Eclipse

Modifying the View of the Screen

- Click on the OpenPerspecive icon lying to the left in the tool line

- Click on the perspective to be modified
- Click on Windows and then Show View in the drop down window
- Lick on Project Explorer
- Click on Other at the bottom of the drop down window
- Click on WindowBuilding at the bottom of the window
- Scroll down and scroll down and click on Structure

Removing Java

https://explainjava.com/uninstall-java-macos/

1. Click on **Launchpad** in the dock
2. Search for **Terminal** and open it.
3. Copy and paste commands to **uninstall Java completely**
    a) sudo rm -rf /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin
    b) sudo rm -rf /Library/PreferencePanes/JavaControlPanel.prefPane
    c) sudo rm -rf /Library/Application\ Support/Oracle/Java/
    d) sudo rm -rf /Library/Java/JavaVirtualMachines

Check what Java versions are available:

ls /Library/Java/JavaVirtualMachines/

Remove the corresponding folder with that version:

sudo rm -fr /Library/Java/JavaVirtualMachines/jdk-9.0.1.jdk

Eclipse App: Eclipse.app

Stored in Macintosh HD/Users/fwesthoff/java-YYYY-MM

JRE Storage Location

Mac: https://docs.oracle.com/javase/10/install/installed-directory-structure-jdk-and-jre.htm#JSJIG-GUID-5CD31FB7-D906-4F4D-BEF6-353FBC68D4F7

Windows: C:\Program Files (x86)\Java\jre n (for 32-bit versions) or C:\Program Files\Java\jre n (for 64-bit versions)

Transfer Project from One Computer to Another: This really did not work well.

https://www.youtube.com/watch?v=H19n5Kho5Io
https://www.youtube.com/watch?v=uVO0_UdQ_Wg

Create the Java Project: Source Computer→ USB

- In your operating system create the folder ESD-USB/ArchiveJarFolder (or whatever folder you prefer)
- In Eclipse's Project Explorer, select the project(s) to be exported
- Click File → Export → General → Archive File → Next
- Browse → Specify the path ESC-USB/ArchiveJarFolder/ArchriveJarFile → Save
- Check the To archive file line. It should be:
  /Volumes/ESC-USB/ArchiveJarFolder/ArchriveJarFile.zip
- Click Finish
- Check that ArchriveJarFile.zip is in the archive folder, ArchiveJarFolder on the USB

Create the project on the destination computer: USB → Destination Computer

- (If desired, select a workspace to place the project(s))
- Click File → Import
- Click General → Existing Project into Workspace→ Next
- Click browser → ArchriveJarFile.zip → Next
- Select the Select Archive File radio button
- The file path to should appear in the textfield and the projects in the text area and be certain that all projects are checked
- Click Finish

Java Workspaces

Create a Workspace on a Computer

- Go to workspace containing the source project
- File → Switch Workspace → Other
- Specify workspace name or enter a new name
- If new workspace click Launch

Deleting a Workspace

- Click on Eclipse (left most entry on the toolbar)
- Click Settings → General→ Start and Shutdown Workspaces
- Click on workspace to be removed→ Click Remove

Odd Behavior

Jar Creation Xml Ant Script

When running the script the jar file is not created. To correct this:

- Right click on your ant script
- Select Run as
- Select External tool configurations...
- Select JRE tab
- Select Run in the same JRE as the workspace

Odd Font Problem

When executing an application a warning was reported in the Console stating the a font was not available. I believe this results from using an old JRE System Library in the Build Path. To correct this:

- Highlight the project
- Select Build Path → Configure Build Path
- Double click JRE System Library
- Expand the Execution environment drop down list and select the most recent JavaSE
- Click finish

Odd Briefcase Behavior

At times the app Briefcase reports that all files are going to be deleted and then recreated. I believe it is caused by a spurious directory on the that appears mysteriously starting with an underscore. Deleting this file appears to solve the problem.