

Series 1: Making File Explorer Software is Easy with Java.....

# GhinExplorer

## Ver 1.0

*Simple File & Directory Explorer*

Volume 1.0



by

Goen-Ghin

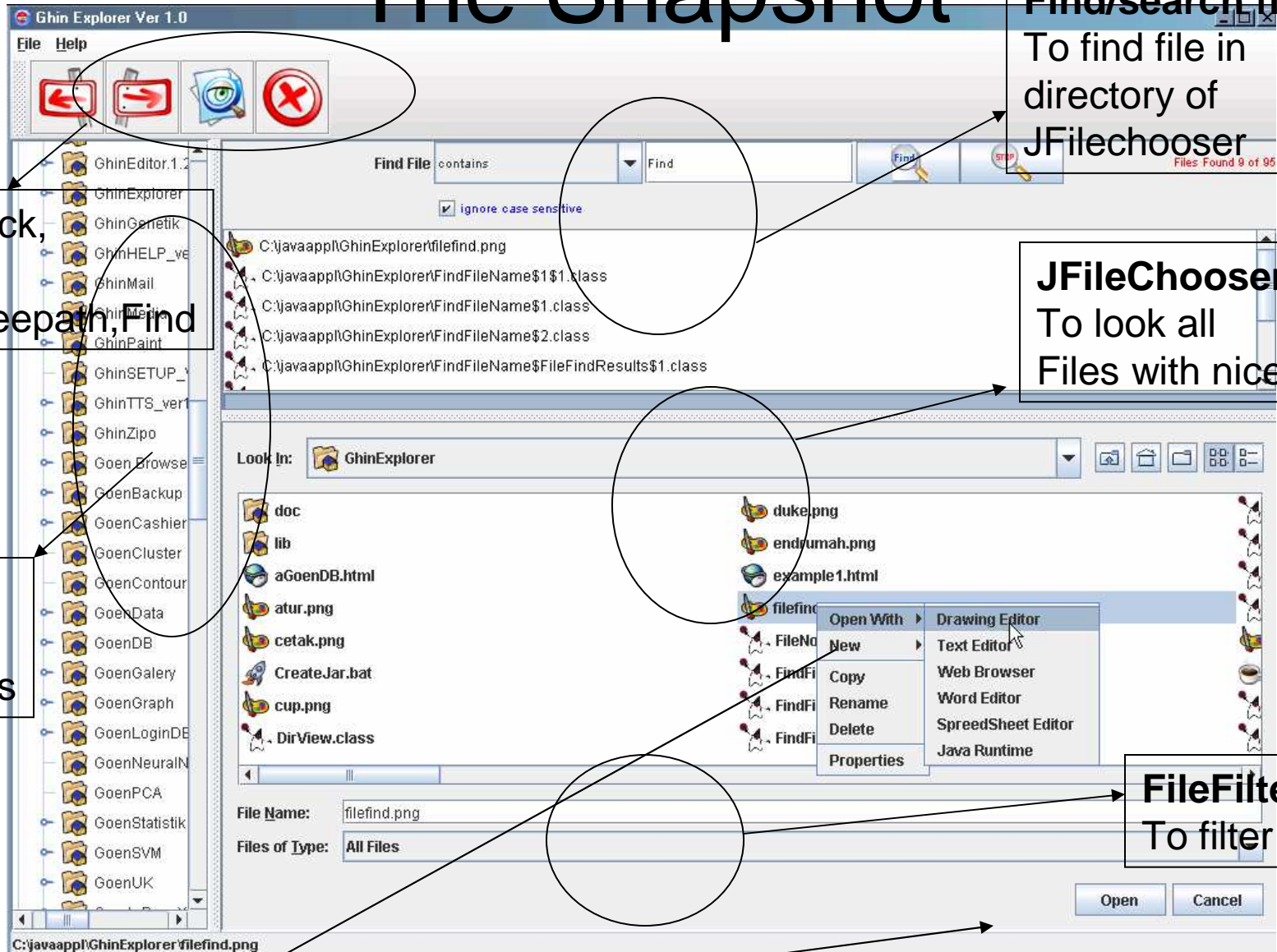
<http://javageo.com>

<http://groups.yahoo.com/group/JavaGeo/>

# Introduction

- This volume 1, will bring you to the world of java programming to create a software which using jswing component
- It's help you to know how to make JTree , JFileChooser to explorer all files and directories
- help you to learn how to find file,copy,delete,rename,paste a file and open with program
- And also give you an idea to design java swing components to any software that you want to create !

# The Snapshot



To Back,  
Next  
Of Jtreepath, Find

JTree  
To list  
All files

Find/search file  
To find file in  
directory of  
JFileChooser

JFileChooser  
To look all  
Files with nice icons

FileFilter  
To filter files

-JPopup and JButton to Choose a file and run with a program

# Class & Functionality

## **Class**

GhinExplorer

FileNode

SimpleFilter

IconCellRenderer

IconData

MyFileChooser

MyToolBar

menulistener

etc

functionalities are

-to Find / Search File

-Stop Find

-Open With ( any program's file)

-Create, Copy, Rename, Delete File

-View file Properties

-Filter file

-Create Directory

-Easy to use

## **Method**

createMenuBar()

getFileNode()

DirExpansionListener()

DirSelectionListener()

getTreeNode()

popup

etc

# Code Using JTree

```
DefaultMutableTreeNode top = new DefaultMutableTreeNode(new IconData(ICON_COMPUTER, null, "Computer"));

DefaultMutableTreeNode node;
File[] roots = File.listRoots();
for (int k=0; k < roots.length; k++)
{
    node = new DefaultMutableTreeNode(new IconData(ICON_DISK, null, new FileNode(roots[k])));
    top.add(node);
node.add( new DefaultMutableTreeNode(new Boolean(true)));
}

myTreeModel = new DefaultTreeModel(top);
myTree = new JTree(myTreeModel);

myTree.putClientProperty("JTree.lineStyle", "Angled");

myTree.addTreeSelectionListener(new TreeSelectionListener()
{
    public void valueChanged(TreeSelectionEvent te)
    {
        try {
            TreePath tp = te.getNewLeadSelectionPath();
            int selRow = myTree.getLeadSelectionRow() ;
            // System.out.println("your selection" +tp.getLastPathComponent());

            String treeSelect = tp.getLastPathComponent().toString();
            System.out.println("your selection" +treeSelect);

        }catch (Exception ex) {}
    }
});

TreeCellRenderer renderer = new IconCellRenderer();
myTree.setCellRenderer(renderer);

myTree.addTreeExpansionListener(new DirExpansionListener());

myTree.addTreeSelectionListener(new DirSelectionListener());

myTree.getSelectionModel().setSelectionMode(
    TreeSelectionMode.SINGLE_TREE_SELECTION);
myTree.setShowsRootHandles(true);
myTree.setEditable(false);
```

# Popup Menu

- `popup = new JPopupMenu();`
- 
- `popup.add(openwithPop = new JMenu("Open With"));`
- `openwithPop.add(openmspaintPop = new JMenuItem("Drawing Editor"));`
- `openwithPop.add(openeditorPop = new JMenuItem("Text Editor"));`
- `openwithPop.add(openbrowserPop = new JMenuItem("Web Browser"));`
- `openwithPop.add(opendocumentPop = new JMenuItem("Word Editor"));`
- `openwithPop.add(openexcelPop = new JMenuItem("SpreadSheet Editor"));`
- `openwithPop.add(openjavaPop = new JMenuItem("Java Runtime"));`
- `openmspaintPop.addActionListener(new menuListener());`
- `openeditorPop.addActionListener(new menuListener());`
- `openbrowserPop.addActionListener(new menuListener());`
- `openmspaintPop.addActionListener(new menuListener());`
- `opendocumentPop.addActionListener(new menuListener());`
- `openjavaPop.addActionListener(new menuListener());`
- `openwithPop.addActionListener(new menuListener());`
- `popup.add(copyPop = new JMenuItem("Copy"));`
- `copyPop.addActionListener(new menuListener());`
- `popup.add(deletePop = new JMenuItem("Delete"));`
- `deletePop.addActionListener(new menuListener());`
- `myFileExplorer.setComponentPopupMenu(popup);`

# Setingan

- //set program to run
- protected static String TEXT\_PROGRAM = "notepad.exe";
- protected static String WEB\_PROGRAM = "firefox.exe";
- protected static String IMAGE\_PROGRAM = "mspaint.exe";
- protected static String SHEET\_PROGRAM = "excell.exe";
- protected static String DOKU\_PROGRAM = "winword.exe";
- protected static String JAVA\_PROGRAM = "java.exe";
- protected static String MEDIA\_PROGRAM = "audioplayer.exe";
- protected static String ZIP\_PROGRAM = "ghinzipo.png";
- protected static String JAR\_PROGRAM = "java.exe -jar";
- protected String[] myfilter = {"zip", "jar", "java", "txt", "exe", "doc", "xls", "html", "png"};
- protected String[] myfiltername = {"ZIP Files", "Java Archive", "Java", "Text File", "Executable", "Document", "Spreadsheets", "HTML File", "Image File"};

//set icon default

```
protected static ImagemIcon EMAK_ICON = new ImagemIcon(ClassLoader.getResource("folder.png"));
protected static ImagemIcon TEXT_ICON = new ImagemIcon(ClassLoader.getResource("kviewshell.png"));
protected static ImagemIcon WEB_ICON = new ImagemIcon(ClassLoader.getResource("mozilla.png"));
protected static ImagemIcon IMAGE_ICON = new ImagemIcon(ClassLoader.getResource("kpaint.png"));
protected static ImagemIcon SHEET_ICON = new ImagemIcon(ClassLoader.getResource("kspread.png"));
protected static ImagemIcon DOKU_ICON = new ImagemIcon(ClassLoader.getResource("kword.png"));
protected static ImagemIcon JAVA_ICON = new ImagemIcon(ClassLoader.getResource("cup.png"));
protected static ImagemIcon MEDIA_ICON = new ImagemIcon(ClassLoader.getResource("multimedia.png"));
protected static ImagemIcon RUN_ICON = new ImagemIcon(ClassLoader.getResource("launcher.png"));
protected static ImagemIcon ZIP_ICON = new ImagemIcon(ClassLoader.getResource("packet.png"));
protected static ImagemIcon CLASS_ICON = new ImagemIcon(ClassLoader.getResource("duke.png"));
```

# Popup Menu and Button Listener

```
• class menuListener implements ActionListener {
•     public void actionPerformed(ActionEvent e) {
•         //run with program that you like
•         if (e.getActionCommand() == "Drawing Editor")
•         {
•             Thread runner1 = new Thread() {
•             public void run() {
•                 try {
•                     runFile(IMAGE_PROGRAM,myFileExplorer.getSelectedFile());
•                 }
•                 catch (Exception e){System.out.println(e);};
•             }
•             };
•             runner1.start();
•         }
•     }
```

```
String stringChoose = ev.getActionCommand();
        if(stringChoose=="ApproveSelection")
        {
if(myFile.endsWith(".txt") ) runFile(TEXT_PROGRAM,file);
if(myFile.endsWith(".xls") ||myFile.endsWith(".odl")) runFile(SHEET_PROGRAM,file);
if(myFile.endsWith(".jpg")||myFile.endsWith(".png") ) runFile(IMAGE_PROGRAM,file);
if(myFile.endsWith(".doc")||myFile.endsWith(".odt") ) runFile(DOKU_PROGRAM,file);
if(myFile.endsWith(".html")||myFile.endsWith(".php") ) runFile(WEB_PROGRAM,file);
if(myFile.endsWith(".java") ) runFile(JAVA_PROGRAM,file);
if(myFile.endsWith(".jar") ) runFile(JAR_PROGRAM,file);
        }
```

# JTree Listener

```
• class DirExpansionListener implements TreeExpansionListener
• {
•     public void treeExpanded(TreeExpansionEvent event)
•     {
•         final DefaultMutableTreeNode node = getTreeNode(event.getPath());
•         final FileNode fnode = getFileNode(node);
•
•         Thread runner = new Thread()
•         {
•             public void run()
•             {
•                 if (fnode != null && fnode.expand(node))
•                 {
•                     Runnable runnable = new Runnable()
•                     {
•                         public void run()
•                         {
•                             myTreeModel.reload(node);
•                         }
•                     };
•                     SwingUtilities.invokeLater(runnable);
•                 }
•             }
•         };
•         runner.start();
•     }
•
•     public void treeCollapsed(TreeExpansionEvent event) {}
• }
•
• class DirSelectionListener implements TreeSelectionListener
• {
•     public void valueChanged(TreeSelectionEvent event)
•     {
•         DefaultMutableTreeNode node = getTreeNode(event.getPath());
•         FileNode fnode = getFileNode(node);
•         if (fnode != null)
•         {
•             m_display.setText(fnode.getFile().getAbsolutePath());
•             myFileExplorer.setCurrentDirectory(new File(m_display.getText()));
•         }
•         else
•             m_display.setText("");
•     }
• }
```

# Source Code

- The completed source code is inside the directory in file of javageo group or
- <http://www.javageo.com>
- You can download the source code...  
Play ,Modify and be succesfull with it...!  
Good luck !!

See you on next volume and series !!