eclipse插件开发常用方法

zjl

2018-10-08T00:00:00.000Z

* [关于UI](#关于ui)
  + [UI 的线程交互](#ui-的线程交互)
  + [在非UI线程中访问UI元素](#在非ui线程中访问ui元素)
  + [在 Composite parent 中添加组件](#在-composite-parent-中添加组件)
    - [一个 Table 分页的 UI Demo（列表和文本框）](#一个-table-分页的-ui-demo列表和文本框)
      * [RemoteSystemInfoView.java](#remotesysteminfoviewjava)
        + [导入包](#导入包)
        + [添加 Table 方式1](#添加-table-方式1)

[添加列表项的监听](#添加列表项的监听)

[添加数据到 table列表](#添加数据到-table列表)

* + - * + [添加 Table布局2](#添加-table布局2)
        + [Group视图](#group视图)

[下拉框](#下拉框)

[styledText](#styledtext)

[按键](#按键)

[文件资源管理器 \*](#文件资源管理器)

[StyledText 显示数据](#styledtext-显示数据)

[文件路径遍历](#文件路径遍历)

* + - * + [Trtee 视图](#trtee-视图)
        + [添加右键菜单 给Table、Tree等继承Control的类都行](#添加右键菜单-给table-tree等继承control的类都行)
        + [注:](#注)
  + [设置默认主题 - 高亮等风格](#设置默认主题-高亮等风格)
  + [自定义对话框 并获取数据](#自定义对话框-并获取数据)
* [关于路径](#关于路径) \* [1、获取工作区路径：](#获取工作区路径) \* [2、获取eclipse安装路径](#获取eclipse安装路径) \* [3、通过文件获取工程Project](#通过文件获取工程project) \* [4、获取workspace](#获取workspace) \* [5、得到整个workspase的根](#得到整个workspase的根) \* [6、获取插件绝对路径](#获取插件绝对路径) \* [7、保存全部](#保存全部) \* [8、通过右键资源菜单获得文件/夹绝对路径](#通过右键资源菜单获得文件夹绝对路径)
* [环境和系统](#环境和系统)
  + [防火墙配置](#防火墙配置)

adadasdad

# 关于UI

## UI 的线程交互

// 和UI的交互  
 static void logInfoViewPut(String msg)  
 {  
 Display display = Display.getDefault();  
 display.asyncExec(new Putlog(msg));  
 }  
   
class Putlog implements Runnable{  
String msg;  
   
public Putlog(String msg) {  
 this.msg = msg;  
}  
@Override  
public void run() {  
 LogInfoView.putLog(msg);  
}  
   
}

## 在非UI线程中访问UI元素

// 设置默认主题  
IWorkbench workbench = PlatformUI.getWorkbench();  
Display dis = workbench.getDisplay();  
   
new Thread() { // 非UI线程  
   
 public void run(){  
   
 dis.syncExec(new Runnable(){ // UI线程  
 @Override  
 public void run() {  
 ThemeDefault themeSet = new ThemeDefault(workbench);  
 themeSet.setTheme(1); // 设置为外观列表里的地2个主题  
 }  
});  
 }  
}.start();

## 在 Composite parent 中添加组件

### 一个 Table 分页的 UI Demo（列表和文本框）

#### RemoteSystemInfoView.java

##### 导入包

import java.io.File;  
import java.text.DecimalFormat;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.Date;  
import java.util.List;  
import org.eclipse.jface.dialogs.MessageDialog;  
import org.eclipse.jface.viewers.TableViewer;  
import org.eclipse.swt.SWT;  
import org.eclipse.swt.custom.CTabFolder;  
import org.eclipse.swt.custom.CTabItem;  
import org.eclipse.swt.custom.SashForm;  
import org.eclipse.swt.custom.StyleRange;  
import org.eclipse.swt.custom.StyledText;  
import org.eclipse.swt.events.MenuEvent;  
import org.eclipse.swt.events.MenuListener;  
import org.eclipse.swt.events.SelectionAdapter;  
import org.eclipse.swt.events.SelectionEvent;  
import org.eclipse.swt.events.SelectionListener;  
import org.eclipse.swt.layout.GridData;  
import org.eclipse.swt.layout.GridLayout;  
import org.eclipse.swt.widgets.Button;  
import org.eclipse.swt.widgets.Combo;  
import org.eclipse.swt.widgets.Composite;  
import org.eclipse.swt.widgets.Control;  
import org.eclipse.swt.widgets.DirectoryDialog;  
import org.eclipse.swt.widgets.Display;  
import org.eclipse.swt.widgets.Event;  
import org.eclipse.swt.widgets.Group;  
import org.eclipse.swt.widgets.Listener;  
import org.eclipse.swt.widgets.Menu;  
import org.eclipse.swt.widgets.MenuItem;  
import org.eclipse.swt.widgets.MessageBox;  
import org.eclipse.swt.widgets.Shell;  
import org.eclipse.swt.widgets.Table;  
import org.eclipse.swt.widgets.TableColumn;  
import org.eclipse.swt.widgets.TableItem;  
import org.eclipse.swt.widgets.Text;  
import org.eclipse.swt.widgets.Tree;  
import org.eclipse.swt.widgets.TreeItem;  
import org.eclipse.ui.part.ViewPart;

public class RemoteSystemInfoView extends ViewPart {  
 // 分页标签的主页面  
 public CTabFolder tabFolder;  
 // 页面子标签项  
 public CTabItem tbtmNewItem;  
 public CTabItem tbtmNewItem\_1;  
 // 子标签视图  
 public TableViewer tableViewer;  
 public TableViewer tableViewer\_1;  
 public TableViewer tableViewer\_2;  
 // 标签组件  
 // 组件 ： 列表  
 public Table table;  
 public Table table\_1;  
 public Table table\_2;  
 // 组件 ： 文本框  
 public Text text;  
   
 // 布局 ：Group，可作为一个容器或页面，添加其他组件、布局等  
 public Group group;  
 // 组件 ： 下拉框  
 public Combo combo;  
 // 组件 ： 按键  
 public Button button;  
 // 组件,和 Text 差不多，和设置字体和颜色  
 public StyledText styledText;  
   
 // 树状图  
 public Tree tree;  
 private TreeItem treeItem;  
 private TreeItem treeItem2;  
 private MenuItem menuItem1;  
 private MenuItem menuItem2;  
   
 // 简单消息对话框  
 MessageBox messagebox;  
   
   
 @Override  
 public void createPartControl(Composite parent) {  
 // 容器初始化 (所有Tab页面都在此容器下)  
 tabFolder = new CTabFolder(parent, SWT.BORDER);  
 tabFolder.setSimple(false);  
   
 // 添加 UI  
 addTableView\_1();  
 addTableView\_2();  
 addGroupView\_3(parent);  
 addTreeView\_4(parent);  
   
 // 添加右键菜单  
 addMenu(table);  
   
 }  
 /\*\*/  
 ~~~ ui 操作 见下文  
 /\*\*/  
 @Override  
 public void setFocus() {  
   
 }  
 void putMsg(String msg)  
 {  
 MessageDialog  
 .openInformation(  
 Display.getCurrent().getActiveShell(),  
 "标题",  
 msg);  
 }  
   
}

##### 添加 Table 方式1

// 添加标签页1 方式1  
// 布局1  
void addTableView\_1()  
{  
 // 标签1  
 tbtmNewItem = new CTabItem(tabFolder, SWT.NONE);  
 tbtmNewItem.setImage(DeviceUtil.getImage("icons/Memory.png"));   
 tbtmNewItem.setText("标签1");  
   
 // 标签的容器，在此容器内添加组件  
 SashForm sashForm = new SashForm(tabFolder, SWT.VERTICAL);  
 tbtmNewItem.setControl(sashForm);  
   
 // (列表视图)标签页面的组件  
 tableViewer = new TableViewer(sashForm, SWT.BORDER | SWT.FULL\_SELECTION);  
 table = tableViewer.getTable();  
 table.setHeaderVisible(true);  
 table.setLinesVisible(true);  
 // 添加列表项的监听  
 addTableListener();  
   
 tableViewer\_2 = new TableViewer(sashForm, SWT.BORDER | SWT.FULL\_SELECTION);  
 table\_2 = tableViewer\_2.getTable();  
 table\_2.setToolTipText("table\_2");  
 table\_2.setHeaderVisible(false);  
 table\_2.setLinesVisible(true);  
 // (文本框视图)标签页面的组件  
 text = new Text(sashForm, SWT.BORDER | SWT.WRAP | SWT.V\_SCROLL | SWT.READ\_ONLY);  
 //text.setFont(DeviceUtil.getFont());设置字体  
 text.setBackground(Display.getCurrent().getSystemColor(SWT.COLOR\_WHITE));  
 text.setEditable(true);  
   
 // sashForm 的权重  
 sashForm.setWeights(new int[] { 161, 170, 103 });  
}

###### 添加列表项的监听

void addTableListener(){  
 table.addListener(SWT.MouseDoubleClick, new Listener() {  
 @Override  
 public void handleEvent(org.eclipse.swt.widgets.Event event) {  
 TableItem tableitm;  
 try {  
 tableitm = table.getSelection()[0];// 获取选中的行  
 putMsg(tableitm.toString());  
   
 } catch (Exception e) {  
 return;  
 }  
 File fitem = (File) tableitm.getData();  
 if (fitem.toString().equals("..")) { //$NON-NLS-1$  
 File file = new File(strlocpath);  
 file = file.getParentFile();  
 listLocalFiles(file, table);  
 return;  
 }  
 if (fitem.isDirectory()) {  
 try {  
 listLocalFiles(fitem, table);  
 } catch (Exception e) {  
 // 遇到无权限的文件夹则返回他上级目录  
 tableitm.toString("无权限的文件夹");  
 File file = fitem.getParentFile();  
 listLocalFiles(file, table);  
 return;  
 }  
 } else {  
 //if (!rimplementation.isListener) {  
 // return;  
 //}  
 //logMessage(Messages.RemoteSystemFtpImplementation\_Startuploading);  
 //rimplementation.Upload(filelist);  
 }  
 }  
 });  
}

###### 添加数据到 table列表

// 添加 数据奥 table 列表  
/\*添加数据到 tab 列表中\*/  
// 给table 添加数据  
public void setTableData\_table(String[] strtitle, String[][] strcontent) {  
 setTableData(table, tableViewer, strcontent, strtitle);  
}  
   
/\*\*  
 \* 给table添加信息公共方法  
 \*  
 \* @param table  
 \* @param tableViewer  
 \*/  
void setTableData(Table table, TableViewer tableViewer, String[][] strcontent,  
 String[] strtitle) {  
 Display.getDefault().syncExec(new Runnable() {  
 @Override  
 public void run() {  
 // 记录table当前位置  
 int iscrollbarlocation = table.getTopIndex();  
 int iSelection = table.getSelectionIndex();  
 // 清空table内容  
 table.removeAll();  
   
 // 重新加载table内容  
 // if(tableViewer.getTable().getColumnCount()==0){  
 TableColumn[] columns = table.getColumns(); // 清除表格原有表头  
 for (TableColumn tc : columns) {  
 tc.dispose();  
 }  
 // 添加表头  
 for (int i = 0; i < strtitle.length; i++) {  
 TableColumn TableContent = new TableColumn(table, SWT.NONE, i);  
 TableContent.setWidth(140);  
 TableContent.setText(strtitle[i]);  
 }  
 // }  
   
 for (int j = 0; j < strcontent.length; j++) {  
 // 添加数据  
 TableItem TableTitle = new TableItem(table, SWT.NONE);  
 String strcont[] = new String[strcontent[j].length];  
 for (int k = 0; k < strcontent[j].length; k++) {  
 strcont[k] = strcontent[j][k];  
 }  
 TableTitle.setText(strcont);  
 }  
 // 设置当前选中信息位置  
 table.setTopIndex(iscrollbarlocation);  
 }  
 });  
}

##### 添加 Table布局2

// 添加标签页2 方式2  
// 布局2  
void addTableView\_2()  
{  
 /\* 方式2 \*/  
 Composite compsoite = new Composite(tabFolder, SWT.NONE);  
 tbtmNewItem\_1 = new CTabItem(tabFolder, SWT.NONE);  
 tbtmNewItem\_1.setImage(DeviceUtil.getImage("icons/Process.png"));  
 tbtmNewItem\_1.setText("标签2");  
 // 组件1  
 tableViewer\_1 = new TableViewer(compsoite, SWT.BORDER | SWT.FULL\_SELECTION);  
 table\_1 = tableViewer\_1.getTable();  
 // 组件布局方式  
 GridData gd\_tablea = new GridData(SWT.FILL, SWT.FILL, true, true, 1, 1);  
 gd\_tablea.heightHint = 220;  
 // 添加布局到 table  
 table\_1.setLayoutData(gd\_tablea);  
 table\_1.setHeaderVisible(true);  
 table\_1.setLinesVisible(true);  
 // 添加UI内容到 CTabItem  
 tbtmNewItem\_1.setControl(compsoite);  
 GridLayout layout = new GridLayout();  
 layout.marginWidth = 0;  
 layout.marginHeight = 0;  
 // 添加布局到 Composite  
 compsoite.setLayout(layout);  
}

##### Group视图

###### 下拉框

###### styledText

###### 按键

###### 文件资源管理器 \*

// 布局3 Group视图 另起一组  
void addGroupView\_3(Composite parent)  
{  
 group = new Group(parent, SWT.NULL);  
 group.setText("Group 布局");  
 GridLayout gridLayout = new GridLayout(2, true);  
 gridLayout.marginHeight = 0;  
 group.setLayout(gridLayout);  
 group.setLayoutData(new GridData(GridData.FILL\_HORIZONTAL));  
   
 /\* 下拉框 \*/  
 combo = new Combo(group, SWT.FLAT);  
 combo.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, true, false, 1, 1));  
 combo.setText("默认数据");  
 combo.add("下拉数据1", 0);//添加到最前  
 combo.add("下拉数据2", 0);//不会覆盖  
 // String[] data = combo.getItems()获得所有项  
 // 为该下拉框选中事件  
 combo.addSelectionListener(new SelectionAdapter() {  
 public void widgetSelected(SelectionEvent e) {  
 String strcomnocontent = combo.getText().trim();// 选择的数据  
 putMsg(strcomnocontent);  
 }  
 });  
   
 /\* 按键 \*/  
 button = new Button(group, SWT.PUSH);  
 button.setImage(DeviceUtil.getImage("icons/up.png"));  
 button.addSelectionListener(new SelectionAdapter() {  
 @Override  
 public void widgetSelected(SelectionEvent e) {  
 // 打开文件资源管理器  
 DirectoryDialog dialog = new DirectoryDialog(parent.getShell());  
 // 返回选中的文件路径  
 String strpath = dialog.open();  
 if (strpath == null) {  
 return;  
 }  
 File file = new File(strpath);  
 if (file.getParentFile() == null) {  
 //buttonUpLocalDir.setEnabled(false);  
 } else {  
 listLocalFiles(file, table);  
 putStyledText(file.toPath().toString());  
 }  
 }  
 });  
 //  
 styledText = new StyledText(group, SWT.BORDER | SWT.V\_SCROLL);  
 styledText.setSize(200, 400);  
 styledText.setEditable(false);  
}

###### StyledText 显示数据

/\*\*  
 \* StyledText 显示数据  
 \* @param message  
 \*/  
public void putStyledText(String message) {  
 styledText.getDisplay().asyncExec(new Runnable() {  
 @Override  
 public void run() {  
 StyleRange styleRange1 = new StyleRange();  
 styleRange1.start = styledText.getCharCount();  
 styleRange1.length = message.length();  
 styleRange1.fontStyle = SWT.BOLD;  
 Shell shell = new Shell();  
 styleRange1.foreground = shell.getDisplay().getSystemColor(SWT.COLOR\_RED);// Display.getCurrent().getActiveShell()  
 styledText.append(message + "\r\n"); //$NON-NLS-1$  
 styledText.setStyleRange(styleRange1);  
 styledText.setSelection(styledText.getCharCount());  
 }  
 });  
}

###### 文件路径遍历

public static long kb = 1024;  
 public static long mb = kb \* 1024;  
 public static long gb = mb \* 1024;  
 public static DecimalFormat df = new DecimalFormat("#,###"); //$NON-NLS-1$  
 String strlocpath = "";// 当前地址  
 /\*  
 \* 在table显示 本地路径 数据  
 \*/  
 private void listLocalFiles(File selDisk, Table table) {  
 if (selDisk == null || selDisk.isFile()) {  
 return;  
 }  
 strlocpath = selDisk.toString();  
 combo.setText(selDisk.toString());// 当前地址  
 table.removeAll();// 移除旧数据  
 File[] listFiles = selDisk.listFiles(); // 获取磁盘文件列表  
 List<File> fileList = new ArrayList<File>();  
 for (File f : listFiles) {  
 fileList.add(f);  
 }  
 Collections.sort(fileList, new Comparator<File>() {  
 @Override  
 public int compare(File o1, File o2) {  
 if (o1.isDirectory() && o2.isFile()) {  
 return -1;  
 }  
 if (o1.isFile() && o2.isDirectory()) {  
 return 1;  
 }  
 return o1.getName().compareTo(o2.getName());  
 }  
 });  
 // 上层目录  
 if (selDisk.getParentFile() != null) {  
   
 File file = new File(".."); //$NON-NLS-1$  
 TableItem itema = new TableItem(table, SWT.NONE);  
 itema.setImage(DeviceUtil.getImage("icons/newfolder.png")); //$NON-NLS-1$  
 itema.setText(new String[] { "..", "", "" }); //$NON-NLS-1$ //$NON-NLS-2$ //$NON-NLS-3$  
 itema.setData(file);  
 } else {  
   
 }  
   
 // 遍历磁盘根文件夹的内容，添加到表格中  
 for (File file : fileList) {  
 TableItem itemsa = new TableItem(table, SWT.NONE);  
 String length = ""; // 获取文件大小 //$NON-NLS-1$  
 long lsize = file.length();  
 if (lsize >= gb) {  
 length = String.format("%.1fGB", (float) lsize / gb) + " (" + df.format(lsize) + ")"; //$NON-NLS-1$ //$NON-NLS-2$ //$NON-NLS-3$  
 } else if (lsize >= mb) {  
 float f = (float) lsize / mb;  
 length = String.format(f > 100 ? "%.0fMB" : "%.1fMB", f) + " (" + df.format(lsize) + ")"; //$NON-NLS-1$ //$NON-NLS-2$ //$NON-NLS-3$ //$NON-NLS-4$  
 } else if (lsize >= kb) {  
 float f = (float) lsize / kb;  
 length = String.format(f > 100 ? "%.0fKB" : "%.1fKB", f) + " (" + df.format(lsize) + ")"; //$NON-NLS-1$ //$NON-NLS-2$ //$NON-NLS-3$ //$NON-NLS-4$  
 } else {  
 length = String.format("%dB", lsize); //$NON-NLS-1$  
 }  
 if (file.isDirectory()) { // 显示文件夹标志  
 length = ""; //$NON-NLS-1$  
 itemsa.setImage(DeviceUtil.getIcon(file));  
 } else {  
 itemsa.setImage(DeviceUtil.getIcon(file));  
 }  
 // 获取文件的最后修改日期  
 SimpleDateFormat myFmt = new SimpleDateFormat(  
 ConstantClass.RemoteSystemFtpView\_DateFormat);  
 String modifDate = myFmt.format(new Date(file.lastModified()));  
 itemsa.setText(new String[] { file.getName(), length, modifDate });  
 itemsa.setData(file);  
 }  
 }

##### Trtee 视图

// 给tab页面添加右键菜单  
// 添加 Tree 视图  
void addTreeView\_4(Composite parent)  
{  
 this.tree = new Tree(parent, SWT.NONE);  
 // 添加双点击事件  
 this.tree.addSelectionListener(new SelectionListener() {  
 // 双击时  
 @Override  
 public void widgetDefaultSelected(SelectionEvent e) {  
 putMsg("TreeView点击响应" + tree.getSelection()[0].getData().toString() );// 获取选中信息  
 }  
   
 // 获得(单击)焦点时  
 @Override  
 public void widgetSelected(SelectionEvent e) {  
 //putMsg("TreeView点击响应widgetSelected");  
 }  
 });  
   
 // 添加右键菜单  
 addMenu(tree);  
   
 treeItem = new TreeItem(tree, SWT.None);  
   
 treeItem.setText("subItem");  
 treeItem.setData("subItemData");  
 treeItem.setImage(DeviceUtil.getImage("icons/connect.png")); //$NON-NLS-1$  
   
 treeItem2 = new TreeItem(treeItem, SWT.None);  
 treeItem2.setText("subItem2");  
 treeItem2.setData("subItemData2");  
 treeItem2.setImage(DeviceUtil.getImage("icons/connect.png")); //$NON-NLS-1$  
   
}

##### 添加右键菜单 给Table、Tree等继承Control的类都行

* 操作项 分割线

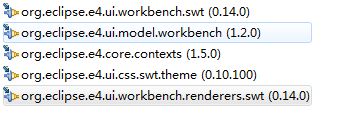
// 添加右键菜单 给Table、Tree等继承Control的类都行  
private void addMenu(Control table) {  
 // 添加右键菜单  
 Menu memorymenu = new Menu(table);  
   
 // 添加操作项  
 MenuItem memoryItem = new MenuItem(memorymenu, SWT.PUSH);  
 // 操作项名称  
 memoryItem.setText(Messages.RemoteSystemInofView\_Refresh);  
 // 添加操作响应监听  
 memoryItem.addListener(SWT.Selection, new Listener() {  
 @Override  
 public void handleEvent(Event event) {  
 Thread memorymenu = new Thread() {  
 public void run() {  
 //操作响应  
 // 列表头  
 String[] strtitle = {"strtitle1","strtitle2"};  
 // 列表数据行  
 String[][] strcontent = {{"strcontent1", "strcontent2"},{"strcontent3", "strcontent4"}};  
 setTableData\_table(strtitle, strcontent);  
 }  
 };  
 memorymenu.setName("memorymenu");  
 memorymenu.start();  
 }  
 });  
   
 // 创建一个分割线  
 MenuItem separator1 = new MenuItem(memorymenu, SWT.SEPARATOR);  
   
 // 添加其他操作项  
 MenuItem connectItem = new MenuItem(memorymenu, SWT.PUSH);  
 connectItem.setText(Messages.RemoteSystemInofView\_Reconnect);  
 connectItem.addListener(SWT.Selection, new Listener() {  
 @Override  
 public void handleEvent(Event event) {  
   
 }  
 });  
   
 // 菜单监听  
 memorymenu.addMenuListener(new MenuListener() {  
 // 菜单显示前  
 @Override  
 public void menuShown(MenuEvent e) {  
 // 设置操作项不可显示  
 memoryItem.setEnabled(true);  
 connectItem.setEnabled(false);  
 // 这里可以设置规则，控制操作项的显示逻辑  
 // \*\*\*  
 }  
   
 // 菜单隐藏后  
 @Override  
 public void menuHidden(MenuEvent e) {  
 }  
 });  
 table.setMenu(memorymenu);  
   
 /\* MenuItem 可重复添加 \*/  
}

##### 注:

* 部分设置图标的语句需要根据实际使用修改

## 设置默认主题 - 高亮等风格

* 需要的导入包



* demo

private IThemeEngine engine;  
 private ITheme currentTheme;  
 private String defaultTheme;  
   
// 调用  
 // 初始化主题  
 IWorkbench workbench = PlatformUI.getWorkbench();  
 init(workbench);  
   
 /\*\*  
 \* 配置默认主题，高对比度  
 \* \*/  
 public void init(IWorkbench workbench) {  
 MApplication application = workbench.getService(MApplication.class);  
 IEclipseContext context = application.getContext();  
 defaultTheme = (String) context.get("org.eclipse.e4.ui.css.theme.e4\_classic");  
 engine = context.get(IThemeEngine.class);  
 engine.setTheme(getCSSThemes(false).get(1), true);  
 }  
   
 /\*\*  
 \* 获取主题  
 \* 主题在 ./plugins/org.eclipse.ui.themes\_1.1.100.v20160411-1921/plugin.xml中声明id  
 \* 主题资源在./plugins/org.eclipse.ui.themes\_1.1.100.v20160411-1921/css中  
 \* @param highContrastMode 是否强对比度  
 \* @return  
 \*/  
 private List<ITheme> getCSSThemes(boolean highContrastMode) {  
 List<ITheme> themes = new ArrayList<>();  
 for (ITheme theme : engine.getThemes()) {  
 if (!highContrastMode && !Util.isGtk()  
 && theme.getId().equals(E4Application.HIGH\_CONTRAST\_THEME\_ID)) {  
 continue;  
 }  
 themes.add(theme);  
 }  
 return themes;  
 }

## 自定义对话框 并获取数据

调用：

···  
UserLoginPage userLoginpage = new UserLoginPage(Display.getCurrent().getActiveShell()  
 , "admin","\*\*\*\*\*\*");  
 if (userLoginpage.open() != Dialog.OK) {  
 return null;  
 }  
   
 userName = userLoginpage.getUser();  
 // isLogin = userLoginpage.isLoginSucess();  
 userState = userLoginpage.getUserState(); // 获取用户状态  
   
···

定义

public class UserLoginPage extends Dialog {  
   
 ···  
   
 public UserLoginPage(Shell parentShell) {  
 super(parentShell);  
 }  
   
   
   
 // UI 布局  
 @Override  
 public Control createDialogArea(Composite parent) {  
 getShell().setText("帐户"); // 标题  
 Composite composite = new Composite(parent, SWT.NONE);  
 GridLayout gl\_composite = new GridLayout(1, false);  
 composite.setLayout(gl\_composite);  
   
 Composite composite\_1 = new Composite(composite, SWT.NONE);  
 composite\_1.setLayoutData(new GridData(SWT.FILL, SWT.FILL, false, false, 1, 1));  
 composite\_1.setLayout(new GridLayout(2, false));  
   
 Label lblNewLabel\_1 = new Label(composite\_1, SWT.NONE);  
 lblNewLabel\_1.setLayoutData(new GridData(SWT.RIGHT, SWT.CENTER, false, false, 1, 1));  
 lblNewLabel\_1.setText("用户名:");  
   
 textUserName = new Text(composite\_1, SWT.BORDER);  
 textUserName.setText(userName);  
 GridData gd\_text = new GridData(SWT.LEFT, SWT.CENTER, false, false, 1, 1);  
 gd\_text.widthHint = 200;  
 textUserName.setLayoutData(gd\_text);  
 textUserName.addVerifyListener(addVerifyListener);  
 textUserName.addModifyListener(new ModifyListener() {  
 public void modifyText(ModifyEvent paramAnonymousModifyEvent) {  
 userName = textUserName.getText().toString();  
 }  
 });  
   
 Label lblnull = new Label(composite\_1, SWT.NONE);  
 Composite composite\_2 = new Composite(composite, SWT.RIGHT);  
 composite\_2.setLayoutData(new GridData(SWT.FILL, SWT.FILL, false, false, 2, 1));  
 composite\_2.setLayout(new GridLayout(3, false));  
   
 // Label lblSinger = new Label(composite\_1, SWT.NONE);  
 Label lblnull2 = new Label(composite\_2, SWT.NONE);  
 lblnull2.setText(" ");// 占位  
 Label lblSinger = new Label(composite\_2, SWT.RIGHT);  
 lblSinger.setText(" ");// 保证合适的label 长度,否则无法正确显示  
 lblSinger.setLayoutData(new GridData(SWT.RIGHT, SWT.CENTER, false, false, 1, 1));  
 URL imageURL = Platform.getBundle(Activator.PLUGIN\_ID).getEntry("icons/ico\_signin.png"); //$NON-NLS-1$  
 ImageDescriptor imgProject = ImageDescriptor.createFromURL(imageURL);  
   
 // 添加背景图片  
 Font font = JFaceResources.getFont("terminal.views.view.font");  
 Image image = new Image(font.getDevice(),imgProject.getImageData());  
 lblSinger.setBackgroundImage(image);  
 lblSinger.addMouseListener(new MouseListener(){  
   
 }  
 ）;  
   
 return parent;  
 }  
   
   
 /\*\*  
 \* @brief Text 改变监听  
 \*/  
 public VerifyListener addVerifyListener = new VerifyListener() {  
 @Override  
 public void verifyText(VerifyEvent event) {  
 // event.doit = false;  
 char myChar = event.character;// 增加的字符  
 // System.out.println("改变监听"+ myChar);  
 }  
 };  
   
 /\*\*  
 \* 重写 ButtonBar 自定义确定和取消文本  
 \*/  
 @Override  
 protected void createButtonsForButtonBar(Composite parent) {  
 // create OK and Cancel buttons by default  
 createButton(parent, IDialogConstants.OK\_ID, "登陆",  
 true);  
 createButton(parent, IDialogConstants.CANCEL\_ID,"取消",  
 false);  
 }  
   
   
 @Override  
 protected void okPressed() {  
 super.okPressed();  
 }  
   
 @Override  
 protected void cancelPressed() {  
 super.cancelPressed();  
 }  
}

# 关于路径

##### 1、获取工作区路径：

Platform.getLocation();   
ResourcesPlugin.getWorkspace();  
Platform.getInstanceLocation();

##### 2、获取eclipse安装路径

Platform.getInstallLocation();

##### 3、通过文件获取工程Project

project pej = ((IFile)o).getProject();  
通过文件获取全路径  
String path = ((IFile)o).getLocation().makeAbsolute().toFile().getAbsolutePath();

##### 4、获取workspace

得到Appliaction workspace;  
FileLocator.toFileURL(PRODUCT\_BUNDLE.getEntry()).getPath(); // error  
得到runtime workspace  
Platform.getInstanceLocation().getURL().getPath();// 工作空间路径  
   
获取工作台  
IWorkbench workbench = PlatformUI.getWorkbench();  
获取工作台窗口  
IWorkbenchWindow window = workbench.getActiveWorkbenchWindow();  
获取工作台页面  
IWorkbenchPage page = window.getActivePage();  
获取当前窗口活动状态的编辑器窗口  
IEditorPart part = page.getActiveEditor();  
获取编辑器窗口输入区  
IEditorInput input = part.getEditorInput();  
获取编辑器的文件,根据Ifile可获得文件路径和文件内容  
if(input instanceof IFileEditorInput){  
IFile file = ((IFileEditorInput)input).getFile();  
}

##### 5、得到整个workspase的根

IWorkspaceRoot root = ResourcesPlugin.getWorkspace().getRoot();  
   
从根查找资源  
IResource resource = root.findMember(new Path("资源文件名"));  
从Bundle查找  
Bundle bundle = Platform.getBundle(pluginId);  
URL fullPathStr = BundleUtility.find(bundle, "filePath");

##### 6、获取插件绝对路径

？？？  
FileLocator.resolve(BuildUIPlugin.getDefault().getBundle().getEntry('/')).getFile();

##### 7、保存全部

try {  
/\*\* 获取工作台打开的的编辑器， 然后保存每个编辑器的内容 \*/  
IEditorReference[] editorReferences = PlatformUI.getWorkbench()  
 .getActiveWorkbenchWindow().getActivePage().getEditorReferences();  
for (IEditorReference iEditorReference : editorReferences) {  
 IEditorPart editor = iEditorReference.getEditor(true);  
 IEditorInput editorInput = iEditorReference.getEditorInput();  
 if (editorInput instanceof IFileEditorInput) {  
 IFileEditorInput fileEditorInput = (IFileEditorInput) editorInput;  
 if (fileEditorInput.getFile().getProject() != null) {  
 if (editor.isDirty()) { /// 对未销毁的进行保存操作  
 editor.doSave(null);  
 }  
 }  
 }  
 }  
} catch (Exception e) {  
   
 }

#### 8、通过右键资源菜单获得文件/夹绝对路径

import java.io.File;  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
   
import org.eclipse.core.commands.AbstractHandler;  
import org.eclipse.core.commands.ExecutionEvent;  
import org.eclipse.core.commands.ExecutionException;  
import org.eclipse.core.resources.IResource;  
import org.eclipse.core.runtime.IPath;  
import org.eclipse.core.runtime.Platform;  
import org.eclipse.jface.text.ITextSelection;  
import org.eclipse.jface.viewers.ISelection;  
import org.eclipse.jface.viewers.IStructuredSelection;  
import org.eclipse.ui.IEditorInput;  
import org.eclipse.ui.IEditorPart;  
import org.eclipse.ui.IFileEditorInput;  
import org.eclipse.ui.IPathEditorInput;  
import org.eclipse.ui.IURIEditorInput;  
import org.eclipse.ui.handlers.HandlerUtil;  
import org.eclipse.cdt.core.model.ICElement;  
import org.eclipse.cdt.core.model.IInclude;  
import org.eclipse.cdt.core.model.IIncludeReference;  
import org.eclipse.cdt.core.model.IMacro;  
import org.eclipse.cdt.internal.ui.cview.IncludeReferenceProxy;  
   
@SuppressWarnings("restriction")  
public class OpenExplorer extends AbstractHandler {  
   
 @SuppressWarnings({ "unchecked", "rawtypes" })  
 @Override  
 public Object execute(ExecutionEvent event) throws ExecutionException {  
 ISelection selection = HandlerUtil.getCurrentSelection(event);  
 List<File> selectedFiles = new ArrayList();  
 if (selection.isEmpty()) {  
 File instanceFile = new File(Platform.getInstanceLocation().getURL().getFile());  
 selectedFiles.add(instanceFile);  
 }  
 if ((selection instanceof IStructuredSelection)) {  
 IStructuredSelection structuredSelection = (IStructuredSelection) selection;  
   
 Iterator<Object> iterator = structuredSelection.iterator();  
 while (iterator.hasNext()) {  
 Object o = iterator.next();  
 if ((o instanceof IResource)) {  
 IResource resource = (IResource) o;  
 File file = new File(resource.getLocationURI());  
 selectedFiles.add(file);  
 } else if ((o instanceof ICElement)) {  
 ICElement cElement = (ICElement) o;  
 File file = new File(cElement.getLocationURI());  
 if ((cElement instanceof IInclude)) {  
 IInclude include = (IInclude) cElement;  
 file = new File(include.getFullFileName());  
 } else if ((cElement instanceof IMacro)) {  
 IMacro macro = (IMacro) cElement;  
 file = new File(macro.getParent().getLocationURI());  
 } else if ((cElement instanceof IIncludeReference)) {  
 IIncludeReference include = (IIncludeReference) cElement;  
 file = new File(include.getPath().toOSString());  
 }  
 selectedFiles.add(file);  
 } else if ((o instanceof IncludeReferenceProxy)) {  
 IncludeReferenceProxy referenceProxy = (IncludeReferenceProxy) o;  
 IPath path = referenceProxy.getReference().getPath();  
 File file = path.toFile();  
 selectedFiles.add(file);  
 }  
 }  
 } else if ((selection instanceof ITextSelection)) {  
 IEditorPart editor = HandlerUtil.getActiveEditor(event);  
 IEditorInput editorInput = editor.getEditorInput();  
 File file = null;  
 if ((editorInput instanceof IFileEditorInput)) {  
 IFileEditorInput input = (IFileEditorInput) editorInput;  
 file = new File(input.getFile().getLocation().toOSString());  
 } else if ((editorInput instanceof IURIEditorInput)) {  
 IURIEditorInput input = (IURIEditorInput) editorInput;  
 file = new File(input.getURI().getPath());  
 } else if ((editorInput instanceof IPathEditorInput)) {  
 IPathEditorInput input = (IPathEditorInput) editorInput;  
 file = new File(input.getPath().toOSString());  
 }  
 if (file != null) {  
 selectedFiles.add(file);  
 }  
 }  
 explorerFiles(selectedFiles);  
 return null;  
 }  
   
 public static void explorerFiles(List<File> files) {  
 Runtime runtime = Runtime.getRuntime();  
 for (File file : files) {  
 String cmd = "explorer "; //$NON-NLS-1$  
 if (file.isFile()) {  
 cmd = cmd + "/select,"; //$NON-NLS-1$  
 }  
 try {  
 String filePath = "\"" + file.getAbsolutePath() + "\""; //$NON-NLS-1$ //$NON-NLS-2$  
 // 这里是路劲出输出  
 System.out.print(filePath);  
 runtime.exec(cmd + filePath);  
 } catch (Exception localException) {  
 }  
 }  
 }  
   
}

# 环境和系统

## 防火墙配置

import java.io.File;  
import java.io.IOException;  
   
import org.eclipse.core.runtime.Platform;  
import org.eclipse.ui.IStartup;  
   
public class Startup implements IStartup {  
   
 // 在防火墙允许程序列表中显示的名字  
 public static final String JAVAW\_FIREWALL\_NAME = "\"Java(TM) Platform SE binary\"";  
 // RealEvo-IDE  
 public static final String IDE\_FIREWALL\_NAME="\"RealEvo-IDE\"";  
   
 @Override  
 public void earlyStartup() {  
 // 获取java安装路径下的javaw.exe文件的绝对路径  
 String javawPath = System.getProperty("java.home") + "/bin/javaw.exe";  
 javawPath = "\"" + javawPath + "\"";  
   
 String idePath = new File(Platform.getInstallLocation().getURL().getPath() + "/"  
 + "RealEvo-IDE.exe").getAbsolutePath();  
 idePath = "\"" + idePath + "\"";  
 try {  
 Runtime runtime = Runtime.getRuntime();  
 // 将java虚拟机从白名单移除  
 runtime.exec("netsh firewall delete allowedprogram " + javawPath);  
 // 将javaw.exe添加到允许列表中  
 runtime.exec("netsh firewall add allowedprogram " + javawPath + " "  
 + JAVAW\_FIREWALL\_NAME + " ENABLE");  
   
 runtime.exec("netsh firewall delete allowedprogram " + idePath);  
 runtime.exec("netsh firewall add allowedprogram " + idePath   
 + " " + IDE\_FIREWALL\_NAME + " ENABLE");  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
   
}