

Forum:[Hack Forums](#)

Thread: [\[Solved - Worked Around\]Swing JTextField DnD\(Drag and Drop\) listener help](#)

??????M1kep?????? - February 4, 2015, 11:35 pm

So I'm trying to figure out how to trigger an actoin in my program whenever somebody drags and drops some text into my text field :/

Currently I tried adding as many listeners as I could and couldn't find anything useful :/

The listeners start at line 195

<https://github.com/M1kep/MangaGoDownloader/blob/DragAndDrop/src/com/m1kep/mangogodownloader/ui/MainUIFrame.java>

Ok so I ended up replacing my original keyListener to a caret Listener which will update whenever the caret moves, seeing as the caret moves when you put drag some text into the text box it will suffice. I'd still love to get an actual way to figure this out though.

I pretty much want to be able to drop text into the field, not delete it from the source, put that text into the field, and display a message in console or prompt just need it to do something after adding the text.

Hol1x - February 5, 2015, 12:40 am

Since i can't post links yet...

Go to Google and search for "java Drag and drop"

Second from top (or just anywere) should say "Drag and Drop and Data Transfer: Examples"

That link contains different examples and i tried "BasicDnD"

Just click "BasicDnD.java" and you will see the full source code.

Hope this help you in someway, it worked for me when i tried using this code.

??????M1kep?????? - February 5, 2015, 10:19 am

(Feb 5, 2015 01:40 AM)jkjklhrew Wrote: [\[CITE\]](#) Since i can't post links yet...

Go to Google and search for "java Drag and drop"

Second from top (or just anywere) should say "Drag and Drop and Data Transfer: Examples"

That link contains different examples and i tried "BasicDnD"

Just click "BasicDnD.java" and you will see the full source code.

Hope this help you in someway, it worked for me when i tried using this code.

Thanks for the reply! I saw that in my research, but it wasn't exactly what I was looking for, I'm pretty sure I've found it though! I'll update when I get home to test it.

Here's what I think it is,

<http://docs.oracle.com/javase/7/docs/api/java/awt/dnd/DropTargetListener.html>

Ok so messed with it a little before work, tried using transfer handlers, and I honestly don't have a clue what I'm doing here's some code that I have so far, it allows you to drag stuff in, but not copy and paste, and I don't know how to get the data from the drag, I think it has something to do with

mime types but I have no fucking clue :/

[DnDTransferHandler.java](#)(Source)

[MainUIFrame.java](#)(Source)(Relevant lines 193 - 203)

??Austin?? - February 5, 2015, 3:31 pm

For other types of data you need to write some more code.

This is the best example I could find:

<http://docs.oracle.com/javase/tutorial/uiswing/examples/dnd/BasicDnDProject/src/dnd/BasicDnD.java>

PHP Code

<?php

/*

* Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

*

* - Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

*

* - Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.

*

* - Neither the name of Oracle or the names of its
* contributors may be used to endorse or promote products derived
* from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

package dnd;

/*

* BasicDnD.java requires no other files.

*/

import java.util.*;

import java.awt.*;

```

import java.awt.event.*;
import java.text.*;
import java.awt.datatransfer.*;

import javax.swing.*;
import javax.swing.text.*;
import javax.swing.tree.*;
import javax.swing.table.*;

public class BasicDnD extends JPanel implements ActionListener {
    private static JFrame frame;
    private JTextArea textArea;
    private JTextField textField;
    private JList list;
    private JTable table;
    private JTree tree;
    private JColorChooser colorChooser;
    private JCheckBox toggleDnD;

    public BasicDnD() {
        super(new BorderLayout());
        JPanel leftPanel = createVerticalBoxPanel();
        JPanel rightPanel = createVerticalBoxPanel();

        //Create a table model.
        DefaultTableModel tm = new DefaultTableModel();
        tm.addColumn("Column 0");
        tm.addColumn("Column 1");
        tm.addColumn("Column 2");
        tm.addColumn("Column 3");
        tm.addRow(new String[]{"Table 00", "Table 01", "Table 02", "Table 03"});
        tm.addRow(new String[]{"Table 10", "Table 11", "Table 12", "Table 13"});
        tm.addRow(new String[]{"Table 20", "Table 21", "Table 22", "Table 23"});
        tm.addRow(new String[]{"Table 30", "Table 31", "Table 32", "Table 33"});

        //LEFT COLUMN
        //Use the table model to create a table.
        table = new JTable(tm);
        leftPanel.add(createPanelForComponent(table, "JTable"));

        //Create a color chooser.
        colorChooser = new JColorChooser();
        leftPanel.add(createPanelForComponent(colorChooser, "JColorChooser"));

        //RIGHT COLUMN
        //Create a textfield.
        textField = new JTextField(30);
        textField.setText("Favorite foods:\nPizza, Moussaka, Pot roast");
        rightPanel.add(createPanelForComponent(textField, "JTextField"));

        //Create a scrolled text area.
        textArea = new JTextArea(5, 30);
        textArea.setText("Favorite shows:\nBuffy, Alias, Angel");
    }

```

```

JScrollPane scrollPane = new JScrollPane(textArea);
rightPanel.add(createPanelForComponent(scrollPane, "JTextArea"));

//Create a list model and a list.
DefaultListModel listModel = new DefaultListModel();
listModel.addElement("Martha Washington");
listModel.addElement("Abigail Adams");
listModel.addElement("Martha Randolph");
listModel.addElement("Dolley Madison");
listModel.addElement("Elizabeth Monroe");
listModel.addElement("Louisa Adams");
listModel.addElement("Emily Donelson");
list = new JList(listModel);
list.setVisibleRowCount(-1);
list.getSelectionModel().setSelectionMode(ListSelectionModel.
MULTIPLE_INTERVAL_SELECTION);

list.setTransferHandler(new TransferHandler() {

    public boolean canImport(TransferHandler.TransferSupport info) {
        // we only import Strings
        if (!info.isDataFlavorSupported(DataFlavor.stringFlavor)) {
            return false;
        }

        JList.DropLocation dl = (JList.DropLocation)info.getDropLocation();
        if (dl.getIndex() == -1) {
            return false;
        }
        return true;
    }

    public boolean importData(TransferHandler.TransferSupport info) {
        if (!info.isDrop()) {
            return false;
        }

        // Check for String flavor
        if (!info.isDataFlavorSupported(DataFlavor.stringFlavor)) {
            displayDropLocation("List doesn't accept a drop of this type.");
            return false;
        }

        JList.DropLocation dl = (JList.DropLocation)info.getDropLocation();
        DefaultListModel listModel = (DefaultListModel)list.getModel();
        int index = dl.getIndex();
        boolean insert = dl.isInsert();
        // Get the current string under the drop.
        String value = (String)listModel.getElementAt(index);

        // Get the string that is being dropped.
        Transferable t = info.getTransferable();
        String data;

```

```

try {
    data = (String)t.getTransferData(DataFlavor.stringFlavor);
}
catch (Exception e) { return false; }

// Display a dialog with the drop information.
String dropValue = "\"" + data + "\" dropped ";
if (dl.isInsert()) {
    if (dl.getIndex() == 0) {
        displayDropLocation(dropValue + "at beginning of list");
    } else if (dl.getIndex() >= list.getModel().getSize()) {
        displayDropLocation(dropValue + "at end of list");
    } else {
        String value1 = (String)list.getModel().getElementAt(dl.getIndex() - 1);
        String value2 = (String)list.getModel().getElementAt(dl.getIndex());
        displayDropLocation(dropValue + "between \"" + value1 + "\" and \"" + value2 + "\"");
    }
} else {
    displayDropLocation(dropValue + "on top of " + "\"" + value + "\"");
}

/** This is commented out for the basicdemo.html tutorial page.
** If you add this code snippet back and delete the
** "return false;" line, the list will accept drops
** of type string.
// Perform the actual import.
if (insert) {
    listModel.add(index, data);
} else {
    listModel.set(index, data);
}
return true;
*/
return false;
}

public int getSourceActions(JComponent c) {
    return COPY;
}

protected Transferable createTransferable(JComponent c) {
    JList list = (JList)c;
    Object[] values = list.getSelectedValues();

    StringBuffer buff = new StringBuffer();

    for (int i = 0; i < values.length; i++) {
        Object val = values[i];
        buff.append(val == null ? "" : val.toString());
        if (i != values.length - 1) {
            buff.append("\n");
        }
    }
}

```

```

return new StringSelection(buff.toString());
}
});
list.setDropMode(DropMode.ON_OR_INSERT);

JScrollPane listView = new JScrollPane(list);
listView.setPreferredSize(new Dimension(300, 100));
rightPanel.add(createPanelForComponent(listView, "JList"));

//Create a tree.
DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode("Mia Familia");
DefaultMutableTreeNode sharon = new DefaultMutableTreeNode("Sharon");
rootNode.add(sharon);
DefaultMutableTreeNode maya = new DefaultMutableTreeNode("Maya");
sharon.add(maya);
DefaultMutableTreeNode anya = new DefaultMutableTreeNode("Anya");
sharon.add(anya);
sharon.add(new DefaultMutableTreeNode("Bongo"));
maya.add(new DefaultMutableTreeNode("Muffin"));
anya.add(new DefaultMutableTreeNode("Winky"));
DefaultTreeModel model = new DefaultTreeModel(rootNode);
tree = new JTree(model);
tree.getSelectionModel().setSelectionMode
(TreeSelectionMode.DISCONTIGUOUS_TREE_SELECTION);
JScrollPane treeView = new JScrollPane(tree);
treeView.setPreferredSize(new Dimension(300, 100));
rightPanel.add(createPanelForComponent(treeView, "JTree"));

//Create the toggle button.
toggleDnD = new JCheckBox("Turn on Drag and Drop");
toggleDnD.setActionCommand("toggleDnD");
toggleDnD.addActionListener(this);

JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
leftPanel, rightPanel);
splitPane.setOneTouchExpandable(true);

add(splitPane, BorderLayout.CENTER);
add(toggleDnD, BorderLayout.PAGE_END);
setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
}

protected JPanel createVerticalBoxPanel() {
    JPanel p = new JPanel();
    p.setLayout(new BoxLayout(p, BoxLayout.PAGE_AXIS));
    p.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
    return p;
}

public JPanel createPanelForComponent(JComponent comp,
String title) {
    JPanel panel = new JPanel(new BorderLayout());
    panel.add(comp, BorderLayout.CENTER);

```

```

    if (title != null) {
        panel.setBorder(BorderFactory.createTitledBorder(title));
    }
    return panel;
}

private void displayDropLocation(final String string) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            JOptionPane.showMessageDialog(null, string);
        }
    });
}

public void actionPerformed(ActionEvent e) {
    if ("toggleDnD".equals(e.getActionCommand())) {
        boolean toggle = toggleDnD.isSelected();
        textArea.setDragEnabled(toggle);
        textField.setDragEnabled(toggle);
        list.setDragEnabled(toggle);
        table.setDragEnabled(toggle);
        tree.setDragEnabled(toggle);
        colorChooser.setDragEnabled(toggle);
    }
}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("BasicDnD");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Create and set up the content pane.
    JComponent newContentPane = new BasicDnD();
    newContentPane.setOpaque(true); //content panes must be opaque
    frame.setContentPane(newContentPane);

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            //Turn off metal's use of bold fonts
            UIManager.put("swing.boldMetal", Boolean.FALSE);
        }
    });
}

```

```

        createAndShowGUI();
    }
    });
}
}

```

??????M1kep?????? - February 6, 2015, 1:12 am

Ok so I have updated the OP with my current work around. :/

(Feb 5, 2015 04:31 PM)?---Abyte ---? Wrote: [\[CITE\]](#) For other types of data you need to write some more code.

This is the best example I could find:

<http://docs.oracle.com/javase/tutorial/uiswing/examples/dnd/BasicDnDProject/src/dnd/BasicDnD.java>

PHP Code

```

<?php
/*
 * Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * - Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 *
 * - Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 *
 * - Neither the name of Oracle or the names of its
 *   contributors may be used to endorse or promote products derived
 *   from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
 * IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

package dnd;

/*
 * BasicDnD.java requires no other files.

```

```
*/
```

```
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.text.*;
import java.awt.datatransfer.*;
```

```
import javax.swing.*;
import javax.swing.text.*;
import javax.swing.tree.*;
import javax.swing.table.*;
```

```
public class BasicDnD extends JPanel implements ActionListener {
    private static JFrame frame;
    private JTextArea textArea;
    private JTextField textField;
    private JList list;
    private JTable table;
    private JTree tree;
    private JColorChooser colorChooser;
    private JCheckBox toggleDnD;
```

```
    public BasicDnD() {
        super(new BorderLayout());
        JPanel leftPanel = createVerticalBoxPanel();
        JPanel rightPanel = createVerticalBoxPanel();
```

```
        //Create a table model.
```

```
        DefaultTableModel tm = new DefaultTableModel();
        tm.addColumn("Column 0");
        tm.addColumn("Column 1");
        tm.addColumn("Column 2");
        tm.addColumn("Column 3");
        tm.addRow(new String[]{"Table 00", "Table 01", "Table 02", "Table 03"});
        tm.addRow(new String[]{"Table 10", "Table 11", "Table 12", "Table 13"});
        tm.addRow(new String[]{"Table 20", "Table 21", "Table 22", "Table 23"});
        tm.addRow(new String[]{"Table 30", "Table 31", "Table 32", "Table 33"});
```

```
        //LEFT COLUMN
```

```
        //Use the table model to create a table.
```

```
        table = new JTable(tm);
        leftPanel.add(createPanelForComponent(table, "JTable"));
```

```
        //Create a color chooser.
```

```
        colorChooser = new JColorChooser();
        leftPanel.add(createPanelForComponent(colorChooser, "JColorChooser"));
```

```
        //RIGHT COLUMN
```

```
        //Create a textfield.
```

```
        textField = new JTextField(30);
        textField.setText("Favorite foods:\nPizza, Moussaka, Pot roast");
        rightPanel.add(createPanelForComponent(textField, "JTextField"));
```

```

//Create a scrolled text area.
textArea = new JTextArea(5, 30);
textArea.setText("Favorite shows:\nBuffy, Alias, Angel");
JScrollPane scrollPane = new JScrollPane(textArea);
rightPanel.add(createPanelForComponent(scrollPane, "JTextArea"));

//Create a list model and a list.
DefaultListModel listModel = new DefaultListModel();
listModel.addElement("Martha Washington");
listModel.addElement("Abigail Adams");
listModel.addElement("Martha Randolph");
listModel.addElement("Dolley Madison");
listModel.addElement("Elizabeth Monroe");
listModel.addElement("Louisa Adams");
listModel.addElement("Emily Donelson");
list = new JList(listModel);
list.setVisibleRowCount(-1);
list.getSelectionModel().setSelectionMode(ListSelectionModel.
MULTIPLE_INTERVAL_SELECTION);

list.setTransferHandler(new TransferHandler() {

    public boolean canImport(TransferHandler.TransferSupport info) {
        // we only import Strings
        if (!info.isDataFlavorSupported(DataFlavor.stringFlavor)) {
            return false;
        }

        JList.DropLocation dl = (JList.DropLocation)info.getDropLocation();
        if (dl.getIndex() == -1) {
            return false;
        }
        return true;
    }

    public boolean importData(TransferHandler.TransferSupport info) {
        if (!info.isDrop()) {
            return false;
        }

        // Check for String flavor
        if (!info.isDataFlavorSupported(DataFlavor.stringFlavor)) {
            displayDropLocation("List doesn't accept a drop of this type.");
            return false;
        }

        JList.DropLocation dl = (JList.DropLocation)info.getDropLocation();
        DefaultListModel listModel = (DefaultListModel)list.getModel();
        int index = dl.getIndex();
        boolean insert = dl.isInsert();
        // Get the current string under the drop.
        String value = (String)listModel.getElementAt(index);

```

```

// Get the string that is being dropped.
Transferable t = info.getTransferable();
String data;
try {
data = (String)t.getTransferData(DataFlavor.stringFlavor);
}
catch (Exception e) { return false; }

// Display a dialog with the drop information.
String dropValue = "\"" + data + "\" dropped ";
if (dl.isInsert()) {
if (dl.getIndex() == 0) {
displayDropLocation(dropValue + "at beginning of list");
} else if (dl.getIndex() >= list.getModel().getSize()) {
displayDropLocation(dropValue + "at end of list");
} else {
String value1 = (String)list.getModel().getElementAt(dl.getIndex() - 1);
String value2 = (String)list.getModel().getElementAt(dl.getIndex());
displayDropLocation(dropValue + "between \"" + value1 + "\" and \"" + value2 + "\"");
}
} else {
displayDropLocation(dropValue + "on top of " + "\"" + value + "\"");
}

/** This is commented out for the basicdemo.html tutorial page.
** If you add this code snippet back and delete the
** "return false;" line, the list will accept drops
** of type string.
// Perform the actual import.
if (insert) {
listModel.add(index, data);
} else {
listModel.set(index, data);
}
return true;
*/
return false;
}

public int getSourceActions(JComponent c) {
return COPY;
}

protected Transferable createTransferable(JComponent c) {
JList list = (JList)c;
Object[] values = list.getSelectedValues();

StringBuffer buff = new StringBuffer();

for (int i = 0; i < values.length; i++) {
Object val = values[i];
buff.append(val == null ? "" : val.toString());
if (i != values.length - 1) {

```

```

buff.append("\n");
}
}
return new StringSelection(buff.toString());
}
});
list.setDropMode(DropMode.ON_OR_INSERT);

JScrollPane listView = new JScrollPane(list);
listView.setPreferredSize(new Dimension(300, 100));
rightPanel.add(createPanelForComponent(listView, "JList"));

//Create a tree.
DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode("Mia Familia");
DefaultMutableTreeNode sharon = new DefaultMutableTreeNode("Sharon");
rootNode.add(sharon);
DefaultMutableTreeNode maya = new DefaultMutableTreeNode("Maya");
sharon.add(maya);
DefaultMutableTreeNode anya = new DefaultMutableTreeNode("Any");
sharon.add(anya);
sharon.add(new DefaultMutableTreeNode("Bongo"));
maya.add(new DefaultMutableTreeNode("Muffin"));
anya.add(new DefaultMutableTreeNode("Winky"));
DefaultTreeModel model = new DefaultTreeModel(rootNode);
tree = new JTree(model);
tree.getSelectionModel().setSelectionMode
(TreeSelectionMode.DISCONTIGUOUS_TREE_SELECTION);
JScrollPane treeView = new JScrollPane(tree);
treeView.setPreferredSize(new Dimension(300, 100));
rightPanel.add(createPanelForComponent(treeView, "JTree"));

//Create the toggle button.
toggleDnD = new JCheckBox("Turn on Drag and Drop");
toggleDnD.setActionCommand("toggleDnD");
toggleDnD.addActionListener(this);

JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
leftPanel, rightPanel);
splitPane.setOneTouchExpandable(true);

add(splitPane, BorderLayout.CENTER);
add(toggleDnD, BorderLayout.PAGE_END);
setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
}

protected JPanel createVerticalBoxPanel() {
    JPanel p = new JPanel();
    p.setLayout(new BoxLayout(p, BoxLayout.PAGE_AXIS));
    p.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
    return p;
}

public JPanel createPanelForComponent(JComponent comp,

```

```

String title) {
    JPanel panel = new JPanel(new BorderLayout());
    panel.add(comp, BorderLayout.CENTER);
    if (title != null) {
        panel.setBorder(BorderFactory.createTitledBorder(title));
    }
    return panel;
}

private void displayDropLocation(final String string) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            JOptionPane.showMessageDialog(null, string);
        }
    });
}

public void actionPerformed(ActionEvent e) {
    if ("toggleDnD".equals(e.getActionCommand())) {
        boolean toggle = toggleDnD.isSelected();
        textArea.setDragEnabled(toggle);
        textField.setDragEnabled(toggle);
        list.setDragEnabled(toggle);
        table.setDragEnabled(toggle);
        tree.setDragEnabled(toggle);
        colorChooser.setDragEnabled(toggle);
    }
}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("BasicDnD");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Create and set up the content pane.
    JComponent newContentPane = new BasicDnD();
    newContentPane.setOpaque(true); //content panes must be opaque
    frame.setContentPane(newContentPane);

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {

```

```

    public void run() {
        //Turn off metal's use of bold fonts
        UIManager.put("swing.boldMetal", Boolean.FALSE);
        createAndShowGUI();
    }
    });
}
}

```

This might help explain what I'm trying to do a littler better.

Me in op Wrote:I pretty much want to be able to drop text into the field, not delete it from the source, put that text into the field, and display a message in console or prompt just need it to do something after adding the text.

(Feb 5, 2015 04:31 PM)?---Abyte ---? Wrote: [\[CITE\]](#) For other types of data you need to write some more code.

This is the best example I could find:

<http://docs.oracle.com/javase/tutorial/uiswing/examples/dnd/BasicDnDProject/src/dnd/BasicDnD.java>

PHP Code

<?php

```

/*
 * Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * - Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 *
 * - Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 *
 * - Neither the name of Oracle or the names of its
 *   contributors may be used to endorse or promote products derived
 *   from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
 * IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```

```

package dnd;

/*
 * BasicDnD.java requires no other files.
 */

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.text.*;
import java.awt.datatransfer.*;

import javax.swing.*;
import javax.swing.text.*;
import javax.swing.tree.*;
import javax.swing.table.*;

public class BasicDnD extends JPanel implements ActionListener {
    private static JFrame frame;
    private JTextArea textArea;
    private JTextField textField;
    private JList list;
    private JTable table;
    private JTree tree;
    private JColorChooser colorChooser;
    private JCheckBox toggleDnD;

    public BasicDnD() {
        super(new BorderLayout());
        JPanel leftPanel = createVerticalBoxPanel();
        JPanel rightPanel = createVerticalBoxPanel();

        //Create a table model.
        DefaultTableModel tm = new DefaultTableModel();
        tm.addColumn("Column 0");
        tm.addColumn("Column 1");
        tm.addColumn("Column 2");
        tm.addColumn("Column 3");
        tm.addRow(new String[]{"Table 00", "Table 01", "Table 02", "Table 03"});
        tm.addRow(new String[]{"Table 10", "Table 11", "Table 12", "Table 13"});
        tm.addRow(new String[]{"Table 20", "Table 21", "Table 22", "Table 23"});
        tm.addRow(new String[]{"Table 30", "Table 31", "Table 32", "Table 33"});

        //LEFT COLUMN
        //Use the table model to create a table.
        table = new JTable(tm);
        leftPanel.add(createPanelForComponent(table, "JTable"));

        //Create a color chooser.
        colorChooser = new JColorChooser();
        leftPanel.add(createPanelForComponent(colorChooser, "JColorChooser"));

        //RIGHT COLUMN

```

```

//Create a textfield.
textField = new JTextField(30);
textField.setText("Favorite foods:\nPizza, Moussaka, Pot roast");
rightPanel.add(createPanelForComponent(textField, "JTextField"));

//Create a scrolled text area.
textArea = new JTextArea(5, 30);
textArea.setText("Favorite shows:\nBuffy, Alias, Angel");
JScrollPane scrollPane = new JScrollPane(textArea);
rightPanel.add(createPanelForComponent(scrollPane, "JTextArea"));

//Create a list model and a list.
DefaultListModel listModel = new DefaultListModel();
listModel.addElement("Martha Washington");
listModel.addElement("Abigail Adams");
listModel.addElement("Martha Randolph");
listModel.addElement("Dolley Madison");
listModel.addElement("Elizabeth Monroe");
listModel.addElement("Louisa Adams");
listModel.addElement("Emily Donelson");
list = new JList(listModel);
list.setVisibleRowCount(-1);
list.getSelectionModel().setSelectionMode(ListSelectionModel.
MULTIPLE_INTERVAL_SELECTION);

list.setTransferHandler(new TransferHandler() {

    public boolean canImport(TransferHandler.TransferSupport info) {
        // we only import Strings
        if (!info.isDataFlavorSupported(DataFlavor.stringFlavor)) {
            return false;
        }

        JList.DropLocation dl = (JList.DropLocation)info.getDropLocation();
        if (dl.getIndex() == -1) {
            return false;
        }
        return true;
    }

    public boolean importData(TransferHandler.TransferSupport info) {
        if (!info.isDrop()) {
            return false;
        }

        // Check for String flavor
        if (!info.isDataFlavorSupported(DataFlavor.stringFlavor)) {
            displayDropLocation("List doesn't accept a drop of this type.");
            return false;
        }

        JList.DropLocation dl = (JList.DropLocation)info.getDropLocation();
        DefaultListModel listModel = (DefaultListModel)list.getModel();

```

```

int index = dl.getIndex();
boolean insert = dl.isInsert();
// Get the current string under the drop.
String value = (String)listModel.getElementAt(index);

// Get the string that is being dropped.
Transferable t = info.getTransferable();
String data;
try {
data = (String)t.getTransferData(DataFlavor.stringFlavor);
}
catch (Exception e) { return false; }

// Display a dialog with the drop information.
String dropValue = "\"" + data + "\" dropped ";
if (dl.isInsert()) {
if (dl.getIndex() == 0) {
displayDropLocation(dropValue + "at beginning of list");
} else if (dl.getIndex() >= list.getModel().getSize()) {
displayDropLocation(dropValue + "at end of list");
} else {
String value1 = (String)list.getModel().getElementAt(dl.getIndex() - 1);
String value2 = (String)list.getModel().getElementAt(dl.getIndex());
displayDropLocation(dropValue + "between \"" + value1 + "\" and \"" + value2 + "\"");
}
} else {
displayDropLocation(dropValue + "on top of " + "\"" + value + "\"");
}

/** This is commented out for the basicdemo.html tutorial page.
** If you add this code snippet back and delete the
** "return false;" line, the list will accept drops
** of type string.
// Perform the actual import.
if (insert) {
listModel.add(index, data);
} else {
listModel.set(index, data);
}
return true;
*/
return false;
}

public int getSourceActions(JComponent c) {
return COPY;
}

protected Transferable createTransferable(JComponent c) {
JList list = (JList)c;
Object[] values = list.getSelectedValues();

StringBuffer buff = new StringBuffer();

```

```

for (int i = 0; i < values.length; i++) {
    Object val = values[i];
    buff.append(val == null ? "" : val.toString());
    if (i != values.length - 1) {
        buff.append("\n");
    }
}
return new StringSelection(buff.toString());
});
list.setDropMode(DropMode.ON_OR_INSERT);

```

```

JScrollPane listView = new JScrollPane(list);
listView.setPreferredSize(new Dimension(300, 100));
rightPanel.add(createPanelForComponent(listView, "JList"));

```

```

//Create a tree.
DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode("Mia Familia");
DefaultMutableTreeNode sharon = new DefaultMutableTreeNode("Sharon");
rootNode.add(sharon);
DefaultMutableTreeNode maya = new DefaultMutableTreeNode("Maya");
sharon.add(maya);
DefaultMutableTreeNode anya = new DefaultMutableTreeNode("Any");
sharon.add(anya);
sharon.add(new DefaultMutableTreeNode("Bongo"));
maya.add(new DefaultMutableTreeNode("Muffin"));
anya.add(new DefaultMutableTreeNode("Winky"));
DefaultTreeModel model = new DefaultTreeModel(rootNode);
tree = new JTree(model);
tree.getSelectionModel().setSelectionMode
(TreeSelectionMode.DISCONTIGUOUS_TREE_SELECTION);
JScrollPane treeView = new JScrollPane(tree);
treeView.setPreferredSize(new Dimension(300, 100));
rightPanel.add(createPanelForComponent(treeView, "JTree"));

```

```

//Create the toggle button.
toggleDnD = new JCheckBox("Turn on Drag and Drop");
toggleDnD.setActionCommand("toggleDnD");
toggleDnD.addActionListener(this);

```

```

JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
leftPanel, rightPanel);
splitPane.setOneTouchExpandable(true);

```

```

add(splitPane, BorderLayout.CENTER);
add(toggleDnD, BorderLayout.PAGE_END);
setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
}

```

```

protected JPanel createVerticalBoxPanel() {
    JPanel p = new JPanel();
    p.setLayout(new BoxLayout(p, BoxLayout.PAGE_AXIS));
    p.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
}

```

```

    return p;
}

public JPanel createPanelForComponent(JComponent comp,
    String title) {
    JPanel panel = new JPanel(new BorderLayout());
    panel.add(comp, BorderLayout.CENTER);
    if (title != null) {
        panel.setBorder(BorderFactory.createTitledBorder(title));
    }
    return panel;
}

private void displayDropLocation(final String string) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            JOptionPane.showMessageDialog(null, string);
        }
    });
}

public void actionPerformed(ActionEvent e) {
    if ("toggleDnD".equals(e.getActionCommand())) {
        boolean toggle = toggleDnD.isSelected();
        textArea.setDragEnabled(toggle);
        textField.setDragEnabled(toggle);
        list.setDragEnabled(toggle);
        table.setDragEnabled(toggle);
        tree.setDragEnabled(toggle);
        colorChooser.setDragEnabled(toggle);
    }
}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("BasicDnD");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Create and set up the content pane.
    JComponent newContentPane = new BasicDnD();
    newContentPane.setOpaque(true); //content panes must be opaque
    frame.setContentPane(newContentPane);

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

```

```
public static void main(String[] args) {  
    //Schedule a job for the event-dispatching thread:  
    //creating and showing this application's GUI.  
    javax.swing.SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            //Turn off metal's use of bold fonts  
            UIManager.put("swing.boldMetal", Boolean.FALSE);  
            createAndShowGUI();  
        }  
    });  
}
```

This might help explain what I'm trying to do a littler better.

Me in op Wrote:I pretty much want to be able to drop text into the field, not delete it from the source, put that text into the field, and display a message in console or prompt just need it to do something after adding the text.