

```

/**
 * Class Name      : EditFrame.java
 * Version         : Build 01.10
 * Date of Change  : 2000 MAR 11
 * Author          : Don Baldwin
 * Copyright       : 2000 ASR Strategic Resources
 * Description     : The GUI and event listeners
 *                  for the edit frame.
 *
 *
 * Version History:
 *
 * Version         Date           Author
 * Build 01.10    2000 MAR 11    Ulf Hedlund
 *
 *                                     Feature enhancement
 *                                     build. Improved GUI.
 *
 * Build 01.00    2000 MAR 10    Anders Nyström
 *
 *                                     Started the class.
 *
 * Known Problems: None
 *
 */

```

```

import java.util.Vector;
import java.util.Enumeration;

```

```

import java.awt.List;
import java.awt.Label;
import java.awt.TextField;
import java.awt.Frame;
import java.awt.TextArea;
import java.awt.Button;
import java.awt.Panel;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.BorderLayout;
import java.awt.PopupMenu;
import java.awt.MenuItem;
import java.awt.Choice;

```

```

import java.awt.event.MouseListener;
import java.awt.event.ItemListener;
import java.awt.event.ActionListener;
import java.awt.event.WindowListener;
import java.awt.event.ActionEvent;
import java.awt.event.WindowEvent;
import java.awt.event.ItemEvent;
import java.awt.event.MouseEvent;

```

```

import java.sql.SQLException;

```

```

/**
 * Edit frame class for AITP NCC Address competition
 */
public class EditFrame extends Frame
{
    /**
     * GUI component declarations
     */
    private TextField    txtInput;
    private TextField    txtName;

    private TextField    fldFirstName;
    private TextField    fldLastName;

```

```

private TextField      fldCompanyName;
private TextArea      areaStreet;
private TextField      fldCity;
private TextField      fldState;
private TextField      fldPostalCode;
private TextField      fldCountryCode;

private Choice         choiceCountry;

private Label          lblFirstName;
private Label          lblLastName;
private Label          lblCompanyName;
private Label          lblStreet;
private Label          lblCity;
private Label          lblState;
private Label          lblPostalCode;
private Label          lblCountryCode;
private Label          lblCountry;
private Label          lblMessage;

private Button         btnEditSave;
private Button         btnEditCancel;

/*
 * Attribute declarations
 */
private static EditFrame objEditFrame = null;
private static boolean isNewAddress = false;
private long longCurrentId = 0;

/*
 * Methods
 */

/**
 * Single instance assessor method for the EditFrame class
 */
public static EditFrame getEditFrame()
{
    if(objEditFrame == null)
    {
        objEditFrame = new EditFrame();
    }

    return objEditFrame;
}

/*
 * The constructor of the class
 */
private EditFrame()
{
    super("Edit Information");

    ActionListener objActionListener;
    WindowListener objWindowListener;
    ItemListener objItemListener;

    /*
     * Adding frame information to the edit frame
     */
    setSize(500,400);
    setLayout(new BorderLayout());

```

```

/*
 * Setting up the edit frame
 */
areaStreet    = new TextArea(10,30);
fldFirstName  = new TextField(30);
fldLastName   = new TextField(30);
fldCompanyName = new TextField(30);
fldCity       = new TextField(30);
fldState      = new TextField(30);
fldPostalCode = new TextField(30);
fldCountryCode = new TextField(30);
choiceCountry = new Choice();

lblFirstName  = new Label("First Name");
lblLastName   = new Label("Last Name");
lblCompanyName = new Label("Company Name");
lblStreet     = new Label("Street Name");
lblCity       = new Label("City Name");
lblState      = new Label("State Name");
lblPostalCode = new Label("Postal Code");
lblCountryCode = new Label("Country Code");
lblCountry    = new Label("Country");

choiceCountry.addItem("USA");
choiceCountry.addItem("Sweden");

fldCountryCode.setEnabled(false);

btnEditSave    = new Button("Save");
btnEditCancel = new Button("Cancel");

Panel pnlNorth  = new Panel();
Panel pnlCenter = new Panel();
Panel pnlSouth  = new Panel();

pnlNorth.setLayout(new GridLayout(3,2));
pnlSouth.setLayout(new GridLayout(7,2));

Panel pnlFirstName = new Panel();
pnlFirstName.setLayout(new GridLayout(1,2));
pnlFirstName.add(lblFirstName);
pnlFirstName.add(fldFirstName);

Panel pnlLastName = new Panel();
pnlLastName.setLayout(new GridLayout(1,2));
pnlLastName.add(lblLastName);
pnlLastName.add(fldLastName);

Panel pnlCompany = new Panel();
pnlCompany.setLayout(new GridLayout(1,2));
pnlCompany.add(lblCompanyName);
pnlCompany.add(fldCompanyName);

pnlNorth.add(pnlFirstName);
pnlNorth.add(pnlLastName);
pnlNorth.add(pnlCompany);

pnlCenter.setLayout(new BorderLayout());
pnlCenter.add("North",lblStreet);
pnlCenter.add("Center",areaStreet);

Panel pnlCity    = new Panel();
pnlCity.setLayout(new GridLayout(1,2));
pnlCity.add(lblCity);
pnlCity.add(fldCity);

```

```

Panel pnlState      = new Panel();
pnlState.setLayout(new GridLayout(1,2));
pnlState.add(lblState);
pnlState.add(fldState);

Panel pnlPostal     = new Panel();
pnlPostal.setLayout(new GridLayout(1,2));
pnlPostal.add(lblPostalCode);
pnlPostal.add(fldPostalCode);

Panel pnlCountryCode = new Panel();
pnlCountryCode.setLayout(new GridLayout(1,2));
pnlCountryCode.add(lblCountryCode);
pnlCountryCode.add(fldCountryCode);

Panel pnlCountry    = new Panel();
pnlCountry.setLayout(new GridLayout(1,2));
pnlCountry.add(lblCountry);
pnlCountry.add(choiceCountry);

Panel pnlStatus     = new Panel();
pnlStatus.setLayout(new GridLayout(1,2));
Label lblStatus = new Label("Status:");
lblMessage = new Label("");
pnlStatus.add(lblStatus);
pnlStatus.add(lblMessage);

Panel pnlEditButtons = new Panel();
pnlEditButtons.setLayout(new GridLayout(1,2));
pnlEditButtons.add(btnEditSave);
pnlEditButtons.add(btnEditCancel);

pnlSouth.add(pnlCity);
pnlSouth.add(pnlState);
pnlSouth.add(pnlPostal);
pnlSouth.add(pnlCountryCode);
pnlSouth.add(pnlCountry);
pnlSouth.add(pnlStatus);
pnlSouth.add(pnlEditButtons);

add("North",pnlNorth);
add("Center",pnlCenter);
add("South",pnlSouth);

/*
 * Add listeners to the different components
 */
objWindowListener = new EditFrameWindowListener();
objActionListener = new EditFrameActionListener();
objItemListener   = new EditFrameItemListener();

addWindowListener(objWindowListener);

btnEditSave.addActionListener(objActionListener);
btnEditCancel.addActionListener(objActionListener);

choiceCountry.addItemListener(objItemListener);
}

/**
 * Display the frame and get focus.
 */
public void showEditFrame()
{

```

```

        setVisible(true);
        focusEditFrame();
    }

/**
 * Hide the frame.
 */
public void hideEditFrame()
{
    MainFrame objFrame;

    objFrame = MainFrame.getMainFrame();

    objFrame.enableMainFrame();
    setVisible(false);
}

/**
 * Get focus on this frame after
 * disabling the main frame.
 */
public void focusEditFrame()
{
    MainFrame objFrame;

    objFrame = MainFrame.getMainFrame();

    objFrame.disableMainFrame();
    requestFocus();
}

/**
 * Get prepared to edit a completely
 * new post.
 */
public void newPost()
{
    fldFirstName.setText("");
    fldLastName.setText("");
    fldCompanyName.setText("");
    fldPostalCode.setText("");
    fldState.setVisible(true);
    fldState.setText("");
    fldCity.setText("");

    areaStreet.setText("");

    choiceCountry.select("USA");
    fldCountryCode.setText("USA");

    lblMessage.setText("");

    isNewAddress = true;
    showEditFrame();
}

/**
 * Get prepared to update/edit an old post.
 */
public void updatePost(Address anObjAddress)
{
    fldFirstName.setText(anObjAddress.getFirstName());
    fldLastName.setText(anObjAddress.getLastName());
    fldCompanyName.setText(anObjAddress.getCompanyName());
    fldCountryCode.setText(anObjAddress.getCountryCode());
}

```

```

fldPostalCode.setText(anObjAddress.getPostalCode());

areaStreet.setText(anObjAddress.getStreet());
choiceCountry.select(anObjAddress.getCountry());

lblMessage.setText("");

// Specialized for USA addresses.
if(anObjAddress.getCountryCode().equals("USA"))
{
    UnitedStatesAddress objUSAddress = (UnitedStatesAddress)anObjAddress;
    fldState.setVisible(true);
    fldState.setText(objUSAddress.getState());
    fldCity.setText(objUSAddress.getCity());
    choiceCountry.select("USA");
}
// Specialized for Swedish addresses.
else
{
    SwedishAddress objSEAddress = (SwedishAddress)anObjAddress;
    fldState.setText("");
    fldState.setVisible(false);
    fldCity.setText(objSEAddress.getCity());
    choiceCountry.select("Sweden");
}

isNewAddress = false;
longCurrentId = anObjAddress.getId();
showEditFrame();
}

/**
 * Listener classes.
 */
private class EditFrameWindowListener implements WindowListener
{
    /**
     * Dummy methods to legally implement the interface.
     */

    /**
     * Dummy method
     */
    public void windowOpened(WindowEvent e)
    {
    }

    /**
     * Dummy method
     */
    public void windowClosed(WindowEvent e)
    {
    }

    /**
     * Dummy method
     */
    public void windowIconified(WindowEvent e)
    {
    }

    /**
     * Dummy method
     */
}

```

```

public void windowDeiconified(WindowEvent e)
{
}

/**
 * Dummy method
 */
public void windowActivated(WindowEvent e)
{
}

/**
 * Dummy method
 */
public void windowDeactivated(WindowEvent e)
{
}

/**
 * Method that handles the event of closing the window.
 */
public void windowClosing(WindowEvent e)
{
    hideEditFrame();
}
}

private class EditFrameItemListener implements ItemListener
{
    // Did we change item in the country choice list?
    public void itemStateChanged(ItemEvent evtItem)
    {
        Object    objSource;

        objSource = evtItem.getSource();

        // This should always be the case.
        if(objSource == choiceCountry)
        {
            String  strItem;

            strItem = choiceCountry.getSelectedItem();

            // USA addresses: show state field, set USA
            // in the country code field.
            if(strItem.equals("USA"))
            {
                fldCountryCode.setText("USA");
                fldState.setVisible(true);
            }
            // Swedish addresses: hide and clear the state
            // field, set SE in the country code field.
            else
            {
                fldCountryCode.setText("SE");
                fldState.setText("");
                fldState.setVisible(false);
            }
        } // END IF inner
    } // END IF outer
}

public class EditFrameActionListener implements ActionListener

```

```

{
/**
 * A method that handles all the interaction with the mouse
 */
public void actionPerformed(ActionEvent evtAction)
{
    Object    objSource;

    objSource = evtAction.getSource();

    /**
     * The user clicks on the cancel button.
     */
    if(objSource == btnEditCancel)
    {
        MainFrame objMainFrame;
        objMainFrame = MainFrame.getMainFrame();

        objMainFrame.refreshMainFrame();
        hideEditFrame();
    }

    /**
     * The user click on the save button in the edit window
     */
    else if(objSource == btnEditSave)
    {
        MainFrame objMainFrame;
        Address    objAddress;

        // Give the user a message that we're saving.
        lblMessage.setText("Saving...");

        // Try to create an address object based on
        // the information in the text fields.
        try
        {
            objAddress = createAddress();
        }
        catch(IllegalArgumentException excIA)
        {
            lblMessage.setText(excIA.getMessage());
            return;
        }

        // If it's new, let the address object handle its own
        // database writing.
        if(isNewAddress == true)
        {
            try
            {
                objAddress.writeAddress();
            }
            catch(SQLException excSQL)
            {
                System.out.println("Exception creating new address: "+excSQL);
            }
        }
        // Else we let the address class delete the address,
        // then make the address write itself to the DB.
        else
        {
            try
            {

```

```

        objAddress.setId(longCurrentId);
        Address.updateAddress(objAddress);
        Registry.updateAddress(objAddress);
    }
    catch(SQLException excSQL)
    {
        System.out.println("Exception updating address: "+excSQL);
    }
}

// Refresh the main window, read all addresses into it
Registry.readAllAddresses();
objMainFrame = MainFrame.getMainFrame();
objMainFrame.refreshMainFrame();
objMainFrame.fetchInformation();

// Hide us, we're done.
hideEditFrame();
}
}

/**
 * Method that tries to create an address object
 * from the information entered in the fields.
 */
public Address createAddress()
    throws IllegalArgumentException
    {

        int        intLength;

        String    strTempFirstName;
        String    strTempLastName;
        String    strTempCompanyName;
        String    strTempStreet;
        String    strTempPostalCode;
        String    strTempCity;
        String    strTempCountryCode;
        String    strTempCountry;
        String    strTempState;

        Address objAddress    = null;

        // Retrieving values from the input fields
        strTempFirstName    = fldFirstName.getText().trim();
        strTempLastName     = fldLastName.getText().trim();
        strTempCompanyName  = fldCompanyName.getText().trim();
        strTempStreet       = areaStreet.getText().trim();
        strTempPostalCode   = fldPostalCode.getText().trim();
        strTempCity         = fldCity.getText().trim();
        strTempCountryCode  = fldCountryCode.getText().trim();
        strTempCountry      = choiceCountry.getSelectedItem().trim();
        strTempState        = fldState.getText().trim();

        intLength    = strTempStreet.length();

        if(intLength < 1)
        {
            throw new IllegalArgumentException("The street name cannot be empty");
        }

        // Sometimes the TextArea adds a \n to the end of the saved string,
        // and we don't really want that. If the last char is a \n, remove it.
        if(strTempStreet.charAt(intLength-1) == '\n')

```

```

    {
        strTempStreet = strTempStreet.substring(0,intLength-1);
    }

// Attempt to construct an Address with the values
if(strTempCountryCode.equals("USA"))
{
    UnitedStatesAddress objUSAddress;
    int                intIndex;
    Vector              vecStreets;

    objUSAddress = new UnitedStatesAddress();

    objUSAddress.setFirstName(strTempFirstName);
    objUSAddress.setLastName(strTempLastName);
    objUSAddress.setCompanyName(strTempCompanyName);
    objUSAddress.setPostalCode(strTempPostalCode);
    objUSAddress.setCity(strTempCity);
    objUSAddress.setState(strTempState);

// Create a vector for the USA street format.
intIndex = strTempStreet.indexOf('\n');
vecStreets = new Vector();

while(intIndex > -1)
    {
        String strPartialStreet;

        strPartialStreet = strTempStreet.substring(0,intIndex);
        strTempStreet = strTempStreet.substring(intIndex+1);
        intIndex = strTempStreet.indexOf('\n');

        vecStreets.addElement(strPartialStreet);
    }

    vecStreets.addElement(strTempStreet);
    objUSAddress.setStreet(vecStreets);

    objAddress = objUSAddress;
}
else
{
    SwedishAddress objSEAddress;
    int                intIndex;
    Vector              vecStreets;

    objSEAddress = new SwedishAddress();

    objSEAddress.setFirstName(strTempFirstName);
    objSEAddress.setLastName(strTempLastName);
    objSEAddress.setCompanyName(strTempCompanyName);
    objSEAddress.setPostalCode(strTempPostalCode);
    objSEAddress.setCountryCode(strTempCountryCode);
    objSEAddress.setCountry(strTempCountry);
    objSEAddress.setCity(strTempCity);
    objSEAddress.setStreet(strTempStreet);

    objAddress = objSEAddress;
}

return objAddress;
}

} // END EditFrame.java

```