

```

Loading vc-cvs...
1  package caltool.schedule_ui;
2
3  import caltool.schedule.*;
4  import mvp.*;
5  import java.util.*;
6  import javax.swing.*;
7  import java.awt.*;
8  import java.awt.event.*;
9
10 /****
11  *
12  * Class RemindInfoSubDialog is a component of the ScheduleAppointment,
13  * ConfirmMeeting, and ScheduleTask Dialogs. The companion model is
14  * RemindInfo. RemindInfoSubDialog displays the row of remind information.
15  */
16 public class RemindInfoSubdialog extends View {
17
18     /**
19     * Construct this with a parent view. Since this is a subview, it need not
20     * have its own model, since it only handles view details. The parent
21     * dialogs for appointments, meetings, and tasks communicate with the
22     * model.
23     */
24     public RemindInfoSubdialog(Screen screen,
25         ScheduleAppointmentDialog parentView) {
26         super(screen, null);
27         this.parentView = parentView;
28
29         maxHeight = 1.9;
30         maxWidth = 1200;
31     }
32
33     /**
34     * Compose this as the default configuration consisting of a remind
35     * checkbox, a time-before text field, a time-before combo box, and a
36     * remind-location combo box.
37     */
38     public Component compose() {
39
40         Box hbox = Box.createHorizontalBox();
41
42         JLabel remindLabel = new JLabel("Remind? ");
43         remindLabel.setForeground(Color.black);
44         remindCheckBox = new JCheckBox();
45         remindCheckBox.addActionListener(new RemindCheckBoxListener(this));
46
47         remindTimeTextField = new JTextField(3);
48         remindTimeTextField.setMaximumSize(
49             new Dimension(3 * remindTimeTextField.getFont().getSize(),
50                 (int)(maxComponentHeight *
51                     remindTimeTextField.getFont().getSize())));
52
53         remindTimeTextField.setEnabled(false);
54         String[] selections =
55             {"minutes before", "hours before", "days before"};
56
57         remindTimeComboBox = new JComboBox(selections);
58         remindTimeComboBox.setEnabled(false);
59         remindTimeComboBox.setMaximumSize(
60             new Dimension(maxComponentWidth, (int)(maxComponentHeight *
61                 remindTimeComboBox.getFont().getSize())));
62
63         String[] selections2 = {"on screen", "email"};
64         remindLocationComboBox = new JComboBox(selections2);
65         remindLocationComboBox.setEnabled(false);
66         remindLocationComboBox.setMaximumSize(
67             new Dimension(maxComponentWidth, (int)(maxComponentHeight *
68                 remindLocationComboBox.getFont().getSize())));
69
70         hbox.add(Box.createHorizontalStrut(15));
71         hbox.add(remindLabel);
72         hbox.add(remindCheckBox);
73         hbox.add(remindTimeTextField);
74         hbox.add(Box.createHorizontalStrut(5));
75         hbox.add(remindTimeComboBox);
76         hbox.add(Box.createHorizontalStrut(15));
77         hbox.add(remindLocationComboBox);
78         hbox.add(Box.createHorizontalStrut(15));
79
80         return hbox;
81     }
82
83     public void update(Observable o, Object arg) {
84
85         RemindInfo remindInfo = (RemindInfo) arg;
86
87         remindCheckBox.setSelected(remindInfo.getIsReminded());
88         remindTimeTextField.setText(Integer.toString(
89             remindInfo.getWhen().getTime()));
90         remindTimeComboBox.setSelectedIndex(remindInfo.getWhen().getTimeUnit().
91             ordinal());
92         remindLocationComboBox.setSelectedIndex(remindInfo.getWhere().
93             ordinal());
94
95     };
96
97     /** The parent appointment dialog view. */
98     ScheduleAppointmentDialog parentView;
99
100    /** The checkbox that indicates whether there's a reminder. */
101    protected JCheckBox remindCheckBox;
102
103    /** The remind time text field. */
104    protected JTextField remindTimeTextField;
105
106    /** The remind time combo box. */
107    protected JComboBox remindTimeComboBox;
108
109    /** The remind location combo box. */
110    protected JComboBox remindLocationComboBox;
111

```

```
112     /** The max height of a text field or combobox; this is necessary since
113         these components stretch when the outer frame is resized, and look very
114         funky when they do */
115     protected final double maxComponentHeight;
116
117     /** The max width of any component; this is only necessary because the max
118         height cannot be set separately, so we must pick some max width. */
119     protected final int maxComponentWidth;
120
121 }
```