

```

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 RecurringInfoSubDialog is a component of the ScheduleAppointment,
13 * ConfirmMeeting, and ScheduleTask Dialogs. The companion model is
14 * RecurringInfo. RecurringInfoSubDialog displays the row of recurring
15 * information in its varying forms, including the monthly details dialog when
16 * appropriate.
17 *
18 * A meeting request dialog has a slightly different version of the recurring
19 * info row. The subclass RecurringInfoMeetingRequestSubDialog displays this
20 * information.
21 *
22 */
23 public class RecurringInfoSubdialog extends View {
24
25     /**
26     * Construct this with a parent view. Since this is a subview, it need not
27     * have its own model, since it only handles view details. The parent
28     * dialogs for appointments, meetings, and tasks communicate with the
29     * model.
30     */
31     public RecurringInfoSubdialog(Screen screen,
32         ScheduleAppointmentDialog parentView) {
33         super(screen, null);
34         this.parentView = parentView;
35
36         maxHeight = 1.9;
37         maxWidth = 1200;
38     }
39
40     /**
41     * Compose this as the default configuration consisting of a recurring
42     * checkbox, a interval combo box, and the weekly interval details.
43     */
44     public Component compose() {
45
46         Box hbox = Box.createHorizontalBox();
47
48         JLabel recurringLabel = new JLabel("Recurring? ");
49         recurringLabel.setForeground(Color.black);
50         recurringCheckBox = new JCheckBox();
51         recurringCheckBox.addActionListener(
52             new RecurringCheckBoxListener(this, parentView));
53
54         intervalLabel = new JLabel("Interval: ");
55         intervalLabel.setForeground(Color.black);
56
57         intervalLabel.setEnabled(false);
58         String[] selections = {"weekly", "biweekly", "monthly", "yearly"};
59         intervalComboBox = new JComboBox(selections);
60         intervalComboBox.setEnabled(false);
61         intervalComboBox.setMaximumSize(
62             new Dimension(maxComponentWidth, (int)(maxComponentHeight *
63                 intervalComboBox.getFont().getSize())));
64
65         hbox.add(Box.createHorizontalStrut(15));
66         hbox.add(recurringLabel);
67         hbox.add(recurringCheckBox);
68         hbox.add(Box.createHorizontalStrut(10));
69         hbox.add(intervalLabel);
70         hbox.add(intervalComboBox);
71         hbox.add(Box.createHorizontalStrut(15));
72         hbox.add(composeWeeklyCheckBoxes());
73         hbox.add(Box.createHorizontalStrut(15));
74
75         return hbox;
76     }
77
78     /**
79     * Compose the seven day-of-the-week checkboxes.
80     */
81     protected Box composeWeeklyCheckBoxes() {
82         Box hbox = Box.createHorizontalBox();
83
84         hbox.add(composeDayCheckBox(sunCheckBox = new JCheckBox(),
85             sunLabel = new JLabel("S")));
86         sunLabel.setForeground(Color.black);
87         hbox.add(composeDayCheckBox(monCheckBox = new JCheckBox(),
88             monLabel = new JLabel("M")));
89         monLabel.setForeground(Color.black);
90         hbox.add(composeDayCheckBox(tueCheckBox = new JCheckBox(),
91             tueLabel = new JLabel("T")));
92         tueLabel.setForeground(Color.black);
93         hbox.add(composeDayCheckBox(wedCheckBox = new JCheckBox(),
94             wedLabel = new JLabel("W")));
95         wedLabel.setForeground(Color.black);
96         hbox.add(composeDayCheckBox(thuCheckBox = new JCheckBox(),
97             thuLabel = new JLabel("T")));
98         thuLabel.setForeground(Color.black);
99         hbox.add(composeDayCheckBox(friCheckBox = new JCheckBox(),
100             friLabel = new JLabel("F")));
101         friLabel.setForeground(Color.black);
102         hbox.add(composeDayCheckBox(satCheckBox = new JCheckBox(),
103             satLabel = new JLabel("S")));
104         satLabel.setForeground(Color.black);
105
106         return hbox;
107     }
108
109     /**
110     * Compose one of the weekly check boxes as a vbox with a label on top and
111     * check box below. I couldn't find any way to have JFC put its built-in

```

```
112     * check box label on top instead of to the right. Whatever.
113     */
114     Box composeDayCheckBox(JCheckBox cbox, JLabel label) {
115         Box vbox = Box.createVerticalBox();
116         label.setAlignmentX(Box.CENTER_ALIGNMENT);
117         label.setEnabled(false);
118         cbox.setAlignmentX(Box.CENTER_ALIGNMENT);
119         cbox.setEnabled(false);
120         vbox.add(label);
121         vbox.add(cbox);
122
123         return vbox;
124     }
125
126     public void update(Observable o, Object arg) {
127         // to be refined
128     }
129
130
131     /** The parent appointment dialog view. */
132     ScheduleAppointmentDialog parentView;
133
134     /** The checkbox that indicates whether the scheduled item recurs. */
135     protected JCheckBox recurringCheckBox;
136
137     /** The interval combobox for weekly, biweekly, monthly, or yearly. */
138     protected JComboBox intervalComboBox;
139
140     /** The label for the interval combobox. It's a data field since it needs
141         to be set enabled or disabled dynamically */
142     protected JLabel intervalLabel;
143
144     /** The Sunday checkbox. */
145     protected JCheckBox sunCheckBox;
146
147     /** The Sunday label. */
148     protected JLabel sunLabel;
149
150     /** The Monday checkbox. */
151     protected JCheckBox monCheckBox;
152
153     /** The Monday label. */
154     protected JLabel monLabel;
155
156     /** The Tuesday checkbox. */
157     protected JCheckBox tueCheckBox;
158
159     /** The Tuesday label. */
160     protected JLabel tueLabel;
161
162     /** The Wednesday checkbox. */
163     protected JCheckBox wedCheckBox;
164
165     /** The Wednesday label. */
166     protected JLabel wedLabel;
167
168     /** The Thursday checkbox. */
169     protected JCheckBox thuCheckBox;
170
171     /** The Thursday label. */
172     protected JLabel thuLabel;
173
174     /** The Friday checkbox. */
175     protected JCheckBox friCheckBox;
176
177     /** The Friday label. */
178     protected JLabel friLabel;
179
180     /** The Saturday checkbox. */
181     protected JCheckBox satCheckBox;
182
183     /** The Saturday label. */
184     protected JLabel satLabel;
185
186     /** The max height of a text field or combobox; this is necessary since
187         these components stretch when the outer frame is resized, and look very
188         funky when they do */
189     protected final double maxComponentHeight;
190
191     /** The max width of any component; this is only necessary because the max
192         height cannot be set separately, so we must pick some max width. */
193     protected final int maxComponentWidth;
194
195 }
```