

```

Loading vc-cvs...
1 package caltool.view_ui;
2
3 import caltool.view.*;
4 import javax.swing.*;
5 import java.awt.*;
6 import java.awt.event.*;
7 // import mvp.*;          // Not imported to avoid conflict with caltool.View
8
9 /**
10 *
11 * Class ViewMenu is the pulldown menu view of the <a href = "View.html" View
12 * </a> model class. The ViewMenu widget is a Java JMenu. Anonymous instances
13 * of JMenuItem are defined for each item in the menu.
14 *
15 */
16 public class ViewMenu extends mvp.View {
17
18     /**
19     * Construct this with the given name as the pulldown label. The given
20     * View is the companion model. Also construct the state-changing menu
21     * items, that must be persistent so they're text can change.
22     */
23     public ViewMenu(mvp.Screen screen, View view, ViewUI viewUI) {
24         super(screen, view);
25         this.viewUI = viewUI;
26
27         showHideAppointmentsItem = new JMenuItem("Hide Appointments");
28         showHideMeetingsItem = new JMenuItem("Hide Meetings");
29         showHideTasksItem = new JMenuItem("Hide Tasks");
30         showHideEventsItem = new JMenuItem("Hide Events");
31     }
32
33     /**
34     * Compose this by inserting each of its menu items into the pulldown menu.
35     * The items are: Item, Day, Week, Month, Year, Next, Previous, Today, Goto
36     * Date, Lists, Filter, Other User, Group, Windows, and Calendars.
37     * Separators are placed after the Year, Goto Date, Filter, and Group
38     * items.
39     *
40     * A menu item is created with the following general code pattern:
41     *
42     *     JMenu.add(new JMenuItem("<em>Item name</em>").addActionListener(
43     *         new ActionListener() {
44     *             public void actionPerformed(ActionEvent e) {
45     *                 <em>model.method()</em>
46     *             }
47     *         }
48     *     );
49     *
50     * Make the widget of this a JMenu.
51     */
52     widget = menu = new JMenu("View");
53
54     /**
55
56         * Add the first group of items -- Item through year.
57         */
58         addItem();
59         addDayItem();
60         addWeekSubmenu();
61         addMonthItem();
62         addYearItem();
63         menu.add(new JSeparator());
64
65     /*
66     * Add the next group of items -- Next through Goto Date.
67     */
68     addNextItem();
69     addPreviousItem();
70     addTodayItem();
71     addGotoDateItem();
72     menu.add(new JSeparator());
73
74     /*
75     * Add the Lists and Filter submenus.
76     */
77     addListsSubmenu();
78     addFilterSubmenu();
79     menu.add(new JSeparator());
80
81     /*
82     * Add the Other User and Group items.
83     */
84     addOtherUserItem();
85     addGroupItem();
86     menu.add(new JSeparator());
87
88     /*
89     * Add Windows and Calendars submenus.
90     */
91     addWindowsSubmenu();
92     addCalendarsSubmenu();
93
94     return widget;
95 }
96
97 /**
98 * Add the 'Item' menu item.
99 */
100 public void addItemItem() {
101     menu.add(new JMenuItem("Item").addActionListener(
102         new ActionListener() {
103             public void actionPerformed(ActionEvent ev) {
104                 viewUI.getItemEditor().update(null,
105                     ((View)model).viewItem());
106                 viewUI.getItemEditor().show();
107             }
108         }
109     ));
110 }
111

```

```

112
113  /**
114   * Add the 'Day' menu item.
115   */
116  public void addDayItem() {
117      menu.add(new JMenuItem("Day")).addActionListener(
118          new ActionListener() {
119              public void actionPerformed(ActionEvent ev) {
120                  System.out.println("In View.viewDay.");
121              }
122          }
123      );
124  }
125  }
126
127  /**
128   * Add the 'Week' menu item, which is a cascading pullright containing
129   * 'Table' and 'List' items.
130   */
131  public void addWeekSubmenu() {
132      submenu = new JMenu("Week");
133      menu.add(submenu);
134
135      /*
136       * Add the 'Week->Table' menu item.
137       */
138      submenu.add(new JMenuItem("Table")).addActionListener(
139          new ActionListener() {
140              public void actionPerformed(ActionEvent ev) {
141                  System.out.println("In View.viewWeekTable.");
142              }
143          }
144      );
145  }
146
147      /*
148       * Add the 'Week->Lists' menu item.
149       */
150      submenu.add(new JMenuItem("Lists")).addActionListener(
151          new ActionListener() {
152              public void actionPerformed(ActionEvent ev) {
153                  System.out.println("In View.viewWeekLists.");
154              }
155          }
156      );
157  }
158  };
159
160  }
161
162  /*
163   * Add the 'Month' menu item.
164   */
165  public void addMonthItem() {
166      menu.add(new JMenuItem("Month")).addActionListener(
167          new ActionListener() {
168
169              public void actionPerformed(ActionEvent ev) {
170                  JFrame jf = new JFrame();
171                  jf.getContentPane().add(new JLabel("xxx"));
172                  jf.setVisible(true);
173                  ((View)model).viewMonth();
174                  viewUI.getMonthlyAgendaDisplay().update(null, null);
175                  viewUI.getMonthlyAgendaDisplay().show();
176              }
177          }
178      );
179  }
180
181  /**
182   * Add the 'Year' menu item.
183   */
184  public void addYearItem() {
185      menu.add(new JMenuItem("Year")).addActionListener(
186          new ActionListener() {
187              public void actionPerformed(ActionEvent ev) {
188                  System.out.println("In View.viewYear.");
189              }
190          }
191      );
192  }
193
194  /**
195   * Add the 'Next' menu item.
196   */
197  public void addNextItem() {
198      menu.add(new JMenuItem("Next")).addActionListener(
199          new ActionListener() {
200              public void actionPerformed(ActionEvent ev) {
201                  System.out.println("In View.gotoNext.");
202              }
203          }
204      );
205  }
206
207  /**
208   * Add the 'Previous' menu item.
209   */
210  public void addPreviousItem() {
211      menu.add(new JMenuItem("Previous")).addActionListener(
212          new ActionListener() {
213              public void actionPerformed(ActionEvent ev) {
214                  System.out.println("In View.gotoPrevious.");
215              }
216          }
217      );
218  }
219
220  /**
221   * Add the 'Today' menu item.
222   */
223  public void addTodayItem() {
224      menu.add(new JMenuItem("Today")).addActionListener(

```

```

224         new ActionListener() {
225             public void actionPerformed(ActionEvent ev) {
226                 System.out.println("In View.gotoToday.");
227             }
228         };
229     };
230 }
231
232 /**
233  * Add the 'Goto Date ...' menu item.
234  */
235 public void addGotoDateItem() {
236     menu.add(new JMenuItem("Goto Date ...").addActionListener(
237         new ActionListener() {
238             public void actionPerformed(ActionEvent ev) {
239                 System.out.println("In View.gotoDate.");
240             }
241         }
242     ));
243 }
244
245 /**
246  * Add the 'Lists' menu item, which is a cascading pullright containing
247  * 'Appointments', 'Meetings', 'Tasks', 'Events', 'All Items', and
248  * 'Custom' items.
249  */
250 public void addListsSubmenu() {
251     submenu = new JMenu("Lists");
252     menu.add(submenu);
253
254     /*
255      * Add the top five items.
256      */
257     addListsAppointmentsItem();
258     addListsMeetingsItem();
259     addListsTasksItem();
260     addListsEventsItem();
261     addListsAllItemsItem();
262     submenu.add(new JSeparator());
263
264     /*
265      * Add the Custom submenu.
266      */
267     addListsCustomSubmenu();
268 }
269
270 /**
271  * Add the 'Lists->Appointments' menu item.
272  */
273 public void addListsAppointmentsItem() {
274     submenu.add(new JMenuItem("Appointments").addActionListener(
275         new ActionListener() {
276             public void actionPerformed(ActionEvent ev) {
277                 viewUI.getAppointmentsListDisplay().update(null, null);
278                 viewUI.getAppointmentsListDisplay().show();
279             }
280         }
281     ));
282
283     /**
284      * Add the 'Lists->Meetings' menu item.
285      */
286     public void addListsMeetingsItem() {
287         submenu.add(new JMenuItem("Meetings").addActionListener(
288             new ActionListener() {
289                 public void actionPerformed(ActionEvent ev) {
290                     System.out.println("In View.viewListsMeetings.");
291                 }
292             }
293         ));
294
295     /**
296      * Add the 'Lists->Tasks' menu item.
297      */
298     public void addListsTasksItem() {
299         submenu.add(new JMenuItem("Tasks").addActionListener(
300             new ActionListener() {
301                 public void actionPerformed(ActionEvent ev) {
302                     System.out.println("In View.viewListsTasks.");
303                 }
304             }
305         ));
306
307     /**
308      * Add the 'Lists->Events' menu item.
309      */
310     public void addListsEventsItem() {
311         submenu.add(new JMenuItem("Events").addActionListener(
312             new ActionListener() {
313                 public void actionPerformed(ActionEvent ev) {
314                     System.out.println("In View.viewListsEvents.");
315                 }
316             }
317         ));
318
319     /**
320      * Add the 'Lists->All Items' menu item.
321      */
322     public void addListsAllItemsItem() {
323         submenu.add(new JMenuItem("All Items").addActionListener(
324             new ActionListener() {
325                 public void actionPerformed(ActionEvent ev) {
326                     System.out.println("In View.viewListsAllItems.");
327                 }
328             }
329         ));
330
331     }
332 }

```

```

336     }
337     );
338 }
339 }
340
341 /**
342  * Add the 'Lists->Custom' submenu.
343  */
344 public void addListsCustomSubmenu() {
345     submenu2 = new JMenu("Custom");
346     submenu.add(submenu2);
347
348     /*
349     * Add a separator.
350     */
351     submenu2.add(new JSeparator());
352
353     /*
354     * Add the 'Lists->Custom->Edit ...' menu item.
355     */
356     submenu2.add(new JMenuItem("Edit ...").addActionListener(
357         new ActionListener() {
358             public void actionPerformed(ActionEvent ev) {
359                 System.out.println("In View.viewListsCustomEdit.");
360             }
361         }
362     ));
363 }
364 }
365
366 /**
367  * Add the 'Filter' menu item, which is a cascading pullright
368  * containing 'Show/Hide Appointments', 'Show/Hide Meetings',
369  * 'Show/Hide Tasks', 'Show/Hide Events', and 'Custom' items.
370  */
371 public void addFilterSubmenu() {
372     submenu = new JMenu("Filter");
373     menu.add(submenu);
374
375     /*
376     * Add the top four items.
377     */
378     addFilterAppointmentsItem();
379     addFilterMeetingsItem();
380     addFilterTasksItem();
381     addFilterEventsItem();
382     submenu.add(new JSeparator());
383
384     /*
385     * Add the Custom submenu.
386     */
387     addFilterCustomSubmenu();
388 }
389
390 /**
391  * Add the 'Filter->Show/Hide Appointments' menu item.

```

```

392  */
393 public void addFilterAppointmentsItem() {
394     submenu.add(showHideAppointmentsItem).addActionListener(
395         new ActionListener() {
396             public void actionPerformed(ActionEvent ev) {
397                 ((View)model).toggleShowHideAppointments();
398                 showHideAppointmentsItem.setText(
399                     (((View)model).areAppointmentsHidden()
400                      ? "Show" : "Hide") + " Appointments");
401             }
402         }
403     );
404 }
405
406 /**
407  * Add the 'Filter->Show/Hide Meetings' menu item.
408  */
409 public void addFilterMeetingsItem() {
410     submenu.add(showHideMeetingsItem).addActionListener(
411         new ActionListener() {
412             public void actionPerformed(ActionEvent ev) {
413                 ((View)model).toggleShowHideMeetings();
414                 showHideMeetingsItem.setText(
415                     (((View)model).areMeetingsHidden()
416                      ? "Show" : "Hide") + " Meetings");
417             }
418         }
419     );
420 }
421
422 /**
423  * Add the 'Filter->Show/Hide Tasks' menu item.
424  */
425 public void addFilterTasksItem() {
426     submenu.add(showHideTasksItem).addActionListener(
427         new ActionListener() {
428             public void actionPerformed(ActionEvent ev) {
429                 ((View)model).toggleShowHideTasks();
430                 showHideTasksItem.setText(
431                     (((View)model).areTasksHidden()
432                      ? "Show" : "Hide") + " Tasks");
433             }
434         }
435     );
436 }
437
438 /**
439  * Add the 'Filter->Show/Hide Events' menu item.
440  */
441 public void addFilterEventsItem() {
442     submenu.add(showHideEventsItem).addActionListener(
443         new ActionListener() {
444             public void actionPerformed(ActionEvent ev) {
445                 ((View)model).toggleShowHideEvents();

```

```

448         showHideEventsItem.setText(
449             ((View)model).areEventsHidden()
450             ? "Show" : "Hide") + " Events");
451     }
452 }
453 );
454 };
455 }
456
457 /**
458  * Add the 'Filter->Custom' submenu.
459  */
460 public void addFilterCustomSubmenu() {
461     submenu2 = new JMenu("Custom");
462     submenu.add(submenu2);
463
464     /*
465      * Add a separator.
466      */
467     submenu2.add(new JSeparator());
468
469     /*
470      * Add the 'Filter->Custom->Edit ...' menu item.
471      */
472     submenu2.add(new JMenuItem("Edit ...").addActionListener(
473         new ActionListener() {
474             public void actionPerformed(ActionEvent ev) {
475                 System.out.println("In View.viewFilterCustomEdit.");
476             }
477         }
478     ));
479 };
480 }
481
482 /**
483  * Add the 'Other User ...' menu item.
484  */
485 public void addOtherUserItem() {
486     menu.add(new JMenuItem("Other User ...").addActionListener(
487         new ActionListener() {
488             public void actionPerformed(ActionEvent ev) {
489                 System.out.println("In View.viewOtherUser.");
490             }
491         }
492     ));
493 }
494
495 /**
496  * Add the 'Group ...' menu item.
497  */
498 public void addGroupItem() {
499     menu.add(new JMenuItem("Group ...").addActionListener(
500         new ActionListener() {
501             public void actionPerformed(ActionEvent ev) {
502                 System.out.println("In View.viewGroup.");
503             }
504         }
505     ));
506 }
507
508 /**
509  * Add the 'Windows' menu item.
510  */
511 public void addWindowsSubmenu() {
512     submenu = new JMenu("Windows");
513     menu.add(submenu);
514
515     /*
516      * Add the Close item with a separator above and below.
517      */
518     submenu.add(new JSeparator());
519     addWindowsCloseItem();
520     submenu.add(new JSeparator());
521
522     /*
523      * Add the Windowing Mode submenu and Magnetize item.
524      */
525     addWindowsWindowingModeSubmenu();
526     addWindowsMagnetizeItem();
527 }
528
529 /**
530  * Add the 'Windows->Close' menu item.
531  */
532 public void addWindowsCloseItem() {
533     submenu.add(new JMenuItem("Close").addActionListener(
534         new ActionListener() {
535             public void actionPerformed(ActionEvent ev) {
536                 System.out.println("In View.close.");
537             }
538         }
539     ));
540 }
541
542 /**
543  * Add the 'Windows->Windowing Mode' submenu.
544  */
545 public void addWindowsWindowingModeSubmenu() {
546     submenu2 = new JMenu("Windowing Mode");
547     submenu.add(submenu2);
548     group = new ButtonGroup();
549
550     /*
551      * Add the three radio button items.
552      */
553     addWindowsTwoWindowModeItem();
554     addWindowsPerLevelModeItem();
555     addWindowsMultiWindowModeItem();
556 }
557
558 /**

```

```

560     * Add the 'Windows->Windowing Mode->Two-Window' menu item.
561     */
562     public void addWindowsTwoWindowModeItem() {
563         submenu2.add(rbItem =
564             new JRadioButtonMenuItem("Two-Window")).addActionListener(
565             new ActionListener() {
566                 public void actionPerformed(ActionEvent ev) {
567                     System.out.println("In View.setTwoWindowMode.");
568                 }
569             });
570     };
571     group.add(rbItem);
572 }
573
574 /**
575  * Add the 'Windows->Windowing Mode->Per-Level' menu item.
576  */
577     public void addWindowsPerLevelModeItem() {
578         submenu2.add(rbItem =
579             new JRadioButtonMenuItem("Per-Level")).addActionListener(
580             new ActionListener() {
581                 public void actionPerformed(ActionEvent ev) {
582                     System.out.println("In View.setPerLevelWindowMode.");
583                 }
584             });
585     };
586     rbItem.setSelected(true);
587     group.add(rbItem);
588 }
589
590 /**
591  * Add the 'Windows->Windowing Mode->Multi-Window' menu item.
592  */
593     public void addWindowsMultiWindowModeItem() {
594         submenu2.add(rbItem =
595             new JRadioButtonMenuItem("Multi-Window")).addActionListener(
596             new ActionListener() {
597                 public void actionPerformed(ActionEvent ev) {
598                     System.out.println("In View.setMultiWindowMode.");
599                 }
600             });
601     };
602     group.add(rbItem);
603 }
604
605 /**
606  * Add the 'Windows->Magnetize' menu item.
607  */
608     public void addWindowsMagnetizeItem() {
609         submenu.add(new JMenuItem("Magnetize")).addActionListener(
610             new ActionListener() {
611                 public void actionPerformed(ActionEvent ev) {
612                     System.out.println("In View.magnetize.");
613                 }
614             });
615     };
616     }
617     };
618 }
619
620 /**
621  * Add the 'Calendars' menu item.
622  */
623     public void addCalendarsSubmenu() {
624         submenu = new JMenu("Calendars");
625         menu.add(submenu);
626     };
627
628     /*
629     * This submenu will be populated by the chronologically-ordered list
630     * of all open calendars.
631     */
632 }
633
634 /** Temp submenu vars, used in item-adding methods */
635 protected JMenuItem submenu, submenu2;
636
637 /** Pre-cast reference to this' widget, which is a menu */
638 protected JMenuItem menu;
639
640 /** Button group for radio button menu items */
641 ButtonGroup group;
642
643 /** Temp radio button menu item */
644 JRadioButtonMenuItem rbItem;
645
646 /** Show/Hide Appointments menu item. It's a persistent data field so it's
647     text can change. */
648 protected JMenuItem showHideAppointmentsItem;
649
650 /** Show/Hide Meetings menu item. It's a persistent data field so it's
651     text can change. */
652 protected JMenuItem showHideMeetingsItem;
653
654 /** Show/Hide Tasks menu item. It's a persistent data field so it's
655     text can change. */
656 protected JMenuItem showHideTasksItem;
657
658 /** Show/Hide Events menu item. It's a persistent data field so it's
659     text can change. */
660 protected JMenuItem showHideEventsItem;
661
662 /** The parent view. */
663 ViewUI viewUI;
664 }

```